1987

An introduction of the theory of nonlinear error-correcting codes

Robert B. Nenno

Rochester Institute of Technology
School of Computer Science and Technology

An Introduction to the Theory of Nonlinear Error-Correcting Codes

by

Robert B. Nenno

A thesis, submitted to
The Faculty of the School of Computer Science and Technology,
in partial fulfillment of the requirements for the degree of
Master of Science in Computer Science.

Approved by:

Professor Donald L. Kreher

Professor Stanislaw P. Radziszowski

Professor James A. Wiseman

October 8, 1987

An Introduction to the Theory of Nonlinear Error-Correcting Codes

a thesis, submitted by

Robert B. Nenno

An Introduction to the Theory of Nonlinear Error-Correcting Codes

Robert B. Nenno

Rochester Institute of Technology

ABSTRACT

Nonlinear error-correcting codes are the topic of this thesis. As a class of codes, it has been investigated far less than the class of linear error-correcting codes. While the latter have many practical advantages, it the former that contain the optimal error-correcting codes. In this project the theory (with illustrative examples) of currently known nonlinear codes is presented. Many definitions and theorems (often with their proofs) are presented thus providing the reader with the opportunity to experience the necessary level of mathematical rigor for good understanding of the subject. Also, the examples will give the reader the additional benefit of seeing how the theory can be put to use. An introduction to a technique for finding new codes via computer search is presented.

Categories and Subject Descriptors: E.4 **[Coding and Information Theory]** Formal models of communication; H.1.1 **[Models and Principles]** Systems and Information Theory

General Terms: Algorithms, Theory

Additional Key Words and Phrases: Bounds on codes, designs, nongroup codes, Hadamard matrices, nonlinear error-correcting codes, uniformly packed codes

October 8, 1987

TABLE OF CONTENTS

CHAPTER 1

Preliminary Ideas

INTRODUCTION

One of the problems that is encountered when transmitting digital data over a communication channel is the occurrence of errors. In order to cope with this several methods of error-correction have been devised [2]. To correct errors an error-correcting code is commonly used. In this thesis we will study nonlinear error-correcting codes.

Definition 1.1

A *nonlinear error-correcting code* is a collection of M codewords (or n-tuples) with components over some alphabet F. We refer to the code as an (n,M,d) code where d is a positive integer such that any two codewords differ in at least d places and d is the largest such number with this property.

Example 1.1

Below is a nonlinear binary (5,4,2) code over $F = \{0,1\}$.

10101
10010
01110
11111

For the sake of completeness we mention, without a formal definition, that a *linear error-correcting code* is the set of all linear combinations of k independent vectors in a vector space V of n-tuples over a field F. Thus, a linear error-correcting code is a vector subspace and it can be given by a basis called the *generator matrix* of the code. We will refer to a linear code using the notation $[n,k,d]$.

In order to provide a brief introduction to how an error-correcting code works let us consider what might happen when 0's or 1's are sent over a communication channel. Consider a message in the form of a string of 0's and 1's that we wish to transmit. Also suppose, for the sake of simplicity, that we only want to transmit the sequences 00, 01, 10, or 11. We might choose to encode these sequences as:

$$\begin{aligned} 00 &= 10101 = c_1 \\ 01 &= 10010 = c_2 \\ 10 &= 01110 = c_3 \\ 11 &= 11111 = c_4 \end{aligned}$$

Let us write $d(x,y)$, where x and y are each n-tuples over some alphabet, to represent the number of positions in which the two n-tuples x and y differ. Formally, we state the following.

Definition 1.2

> The *Hamming distance* between two codewords (n-tuples) is given by $d(x,y)$. The *Hamming weight* of a codeword x is the number of nonzero components of x and is denoted by $\mathrm{wt}(x)$.

Next consider the transmission of some codewords.

Want to Send	Encode and Send	Enroute	Receive	Decode to
01	10010	No error	10010	01
00	10101	Error in bit 3	10001	?

Obviously, as seen in the first row where there are no errors, everything works fine. Consider row two. In order to correct the error in bit 3 in 10001, so that we can correctly decode into 00, we assume that when bits are corrupted to some incorrect value that the most likely situation is that fewer rather that more errors have occurred. This is usually a good assumption and thus we calculate $d(\text{received codeword},c_i)$ for $i = 1, 2, 3, 4$.

$$d(10001,c_1) = 1$$

$$d(10001,c_2) = 2$$

$$d(10001,c_3) = 5$$

$$d(10001,c_4) = 3$$

Clearly, then 10001 is "closest to" c_1. So 10001 is corrected to 10101 and finally we decode 10101 back into 00.

From this one illustration we have seen single-error-correction take place. Schematically, the process is diagramed in figure 1.4.

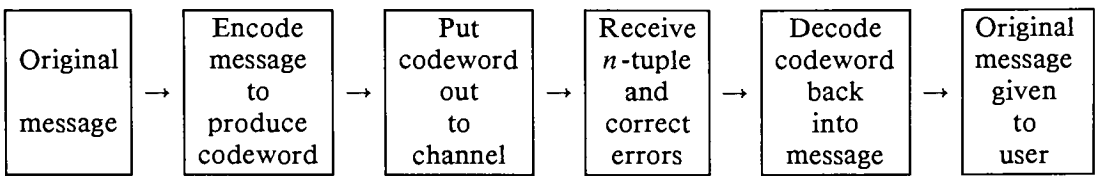


Figure 1.4

It is important to observe that the reason single-error-correction can take place in a code is that the minimum Hamming distance between any two codewords of the code is 3. For if there were two codewords whose Hamming distance were 2 such as with 01111 and 10111, then if 00111 were received there would be no way to know whether bit 1 or bit 2 should be corrected. We state the following definition.

Definition 1.3

The minimum Hamming distance between pairs of distinct codewords in a code is called the *Hamming distance of the code.*

Theorem 1.1 [1]

A code whose minimum Hamming distance is d can correct $t = [\frac{1}{2}(d - 1)]$ errors, where [ ] denotes the greatest integer function. Hence, if a code is t-error-correcting, then $d \geq 2t + 1$.

proof:

The Hamming distance is given as d. Now suppose codeword c is sent and that it is corrupted into c'. If c' is to be decoded to c, it must be the case that $d(c,c') < \frac{d}{2}$. Thus, the number of correctable errors $t < \frac{d}{2}$. If d is even this implies $t < \frac{d}{2} - 1$, whereas if d is odd $t < \frac{d-1}{2}$. Hence, the number of errors that can be corrected is $t = [\frac{1}{2}(d - 1)]$.

It sometimes is possible to correct certain error patterns with t errors even when $d < 2t + 1$, although there is no guarantee of this because it depends upon which codeword is transmitted and on the particular pattern of t errors that occurs in the codeword.

In order to enhance our understanding of all this consider the space of q-ary n-tuples (we depict these as points in n-dimensional space). A code then is some subset of the n-tuples. If d is the Hamming distance of this code and t is the largest integer satisfying $d \geq 2t + 1$ then we can draw nonintersecting spheres of radius t around each codeword. Figure 1.5 shows two such codewords A and B with spheres of radius t.

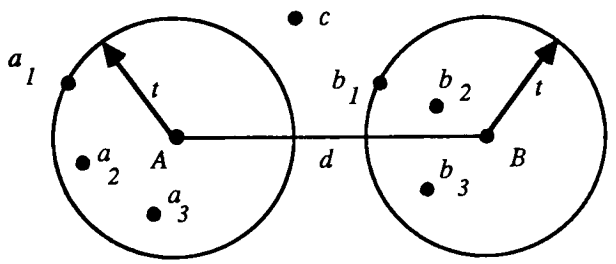


Figure 1.5

Assume $d(a_i,A) \leq t$ and similarly that $d(b_i,B) \leq t$ for $i = 1, 2, 3$. This means that the a_i are n-tuples in the sphere about A and the b_i are n-tuples in the sphere about B. The n-tuple c might (but not necessarily) be in a sphere of radius t about some other codeword. Hence, a received n-tuple in a sphere is corrected to the codeword at the center of that sphere. When t or fewer errors occur, the received n-tuple will be in the proper sphere and will be corrected to the proper codeword. An n-tuple with more than t errors will not lie in its "home sphere", but rather will either:

1) lie in another sphere about some codeword and thus will not be corrected to the proper codeword, or
2) lie in the space between two spheres.

Possibility (2) can be handled as follows. It can be designated as an unrecognizable error pattern and, hence, no attempt at correction is made. The alternative to this is to decode every received codeword to the closest codeword and any n-tuple that is equidistant from two or more closest spheres is arbitrarily assigned to one of those closest spheres. This alternative is called *complete decoding* and assures us that when more than t errors occur we at least occasionally will still be able to find the correct codeword. We mention in passing that when all the spheres of radius t around the codewords are disjoint and collectively contain all the n-tuples the code

is called *perfect.*

An important question about a nonlinear code is whether it is equivalent to another nonlinear code. We settle that query with the definition which follows.

Definition 1.4

Two (n,M,d) binary codes C and D over $F = \{0,1\}$ are *equivalent* if and only if one can be obtained from the other by permuting the n components and performing a component-wise modulo 2 addition of a constant vector to each codeword. That is, C is equivalent to D if and only if there exists a permutation π and a vector a such that $D = \{\pi(u) + a : u \in C\}$, where the + represents component-wise modulo 2 addition.

Example 1.2

Consider the code from example 1.1 which we denote by C.

$$C = \begin{bmatrix} 1\,0\,1\,0\,1 \\ 1\,0\,0\,1\,0 \\ 0\,1\,1\,1\,0 \\ 1\,1\,1\,1\,1 \end{bmatrix}.$$

Now let us define the permutation π which permutes components 1 and 3. Moreover, let $a = 00100$. Then

$$D = \{\pi(u) + a : u \in C\} = \begin{bmatrix} 1\,0\,0\,0\,1 \\ 0\,0\,0\,1\,0 \\ 1\,1\,1\,1\,0 \\ 1\,1\,0\,1\,1 \end{bmatrix}$$

is a code which, by definition, is equivalent to C.

Definition 1.4 can be generalized as follows. Let C and D be (n,M,d) codes over an alphabet of q elements. Then C and D are *equivalent* provided there exist n permutations $\sigma_1, \sigma_2, \ldots, \sigma_n$ of the q elements and a permutation Π of the n coordinate positions such that if $(u_1, \ldots, u_n) \in C$, then $\Pi(\sigma_1(u_1), \ldots, \sigma_n(u_n)) \in D$.

Note by viewing an (n,M,d) code as an $M \times n$ matrix as in the example above, that permuting rows, columns, or the symbols within any column will produce an equivalent code. Furthermore, equivalent codes have the same Hamming distance.

RATE OF A CODE

In the previous section we saw the basic idea of how data is encoded, corrected, and decoded. In particular, we saw a case where 2-bit sequences were encoded to become 5-bit sequences which, of course, were eventually decoded back to 2-bit sequences. Hence, a price

we pay to obtain error-correcting capability is that of transmitting extra digits. In order that we have a way of measuring that price we make the following definition.

Definition 1.5

The *rate* (or *information rate*), R, of a binary (n,M,d) code is given by $R = \dfrac{\log_2 M}{n}$.

In general, an (n,M,d) code over an alphabet F with $|F| = b$, has rate $R = \dfrac{\log_b M}{n}$.

Example 1.3

The rate of the code in example 1.1 is

$$R = \frac{\log_2 M}{n} = \frac{\log_2 4}{5} = \frac{2}{5}.$$

For the reader familiar with binary linear error-correcting codes we point out that $\log_2 M = \log_2 2^k = k$, which is equal to the number of message bits and thus

$$R = \frac{k}{n} = \frac{\text{number of message bits in a transmitted block}}{\text{total number of bits in a transmitted block}}.$$

One of the advantages of a binary nonlinear code over a binary linear code is that for a given n and d, M is often greater than 2^k and thus in these cases we obtain higher information rates with nonlinear codes. Later in this chapter, we investigate the size of M so that we will be able to know more about a particular code.

DISTANCE DISTRIBUTION OF A NONLINEAR CODE

We have seen the importance of the Hamming distance of a code since it is related to the code's error-correcting capability. Furthermore, the weights of codewords in a code often provide sufficient information to aid in the determination of M or d for a nonlinear code.

Definition 1.6

If C is an (n,M,d) code, let A_i = the number of codewords of weight i. The numbers $A_0, A_1, \ldots, A_n$, give the *weight distribution* of C.

Note also that $\sum A_i = M$. For linear codes we mention that it is true that an observer standing at any of the codewords will see A_i codewords at a distance i from the point of observation. However, for nonlinear codes that might not be the case as can be seen in example 1.4.

Example 1.4

Consider the nonlinear code given by

$$C = \begin{bmatrix} 0\,0 \\ 0\,1 \\ 1\,1 \end{bmatrix}.$$

The weight distribution is $A_0 = 1$, $A_1 = 1$, $A_2 = 1$. An observer standing at 00 will see A_i codewords at a distance i from the point of observation. However, an observer standing at 01 will see one codeword at a distance 0 and two codewords at a distance 1 from the point of observation.

If, for a nonlinear code C, the distribution of distances from a given codeword $c_i \in C$ to codewords $c_j \in C$ remains the same, regardless of the choice of c_i, then C is said to be *distance invariant*. Since nonlinear codes are not always distance invariant the following definition is useful.

Definition 1.7

The *distance distribution* of a code C consists of the numbers $B_0, B_1, \ldots, B_n$, where

$$B_i = \frac{1}{M} \times \text{the number of ordered pairs of codewords } u, v \text{ such that } d(u,v) = i.$$

For linear codes the distance distribution is identical to the weight distribution, but for nonlinear codes that is not necessarily the case as the next example verifies.

Example 1.5

Consider again the code C from example 1.4 which was seen to be not distance invariant. We now calculate the distance distribution.

$B_0 = \frac{1}{3} \times 3 = 1$. (Since $M = 3$ and there are three ordered pairs of codewords such that the distance between members of any pair is 0, i.e. (00,00), (01,01), and (11,11).)

$B_1 = \frac{1}{3} \times 4 = \frac{4}{3}$. (Since $M = 3$ and there are four ordered pair of codewords such that the distance between members of any pair is 1, i.e. (00,01), (01,00), (01,11), and (11,01).)

$B_2 = \frac{1}{3} \times 2 = \frac{2}{3}$. (Since $M = 3$ and there are two ordered pairs of codewords such that the distance between members of any pair is 2, i.e. (00,11), and (11,00).)

Thus it can be seen that the definition of distance distribution produces the numbers B_i which are average distances.

AN UPPER BOUND ON M

We next prove the following theorem which gives an upper bound on M known as the *Plotkin bound.*

Theorem 1.2 [3]

For any (n,M,d) code C for which $n < 2d$,

$$M \leq 2\left\lfloor \frac{d}{2d-n} \right\rfloor,$$

where $\lfloor\ \rfloor$ denotes the greatest integer function.

proof:

For any two distinct codewords u, v we have $d(u,v) \geq d$. Now form the sum, S, of all possible such distances between distinct pairs of codewords.

$$S = \sum_{u \in C} \sum_{v \in C} d(u,v) \geq 2\binom{M}{2} d.$$

$$S \geq M(M-1)d. \tag{1.1}$$

Now consider the $M \times n$ matrix of codewords and let the ith column contain x_i 0's and $M - x_i$ 1's. This one column contributes $2x_i(M - x_i)$ to S. Hence,

$$S = \sum_{i=1}^{n} 2x_i(M - x_i). \tag{1.2}$$

However, from elementary calculus it is known that the product $x(M-x)$ is maximized when $x = \frac{1}{2}M$. Thus, if M is even we have:

$$S \leq \sum_{i=1}^{n} 2\left(\frac{M}{2}\right)\left(\frac{M}{2}\right) = \frac{nM^2}{2}. \tag{1.3}$$

So by combining inequalities (1.1) and (1.3),

$$M(M-1)d \leq \frac{nM^2}{2}.$$

Whence,

$$M \leq \frac{2d}{2d-n}.$$

Now since M is even,

$$M \leq 2\left\lfloor \frac{d}{2d-n} \right\rfloor.$$

Next consider what happens if M is odd. In that case S is no longer maximized by $x = \frac{1}{2}M$. Rather, x must be equal to either $\frac{1}{2}(M-1)$ or $\frac{1}{2}(M+1)$. Substituting either of these values into (1.2) we obtain:

$$S \leq \frac{n(M^2-1)}{2}. \tag{1.4}$$

Then by combining (1.1) and (1.4) we have:

$$M(M-1)d \leq \frac{n(M^2-1)}{2}.$$

Whence,

$$M \leq \frac{2d}{2d-n} - 1.$$

Thus for M odd we also find:

$$M \leq 2\left\lfloor \frac{d}{2d-n} \right\rfloor.$$

Example 1.6

For a nonlinear code with $n = 16$ and $d = 9$ we have $n < 2d$ and thus by theorem 1.2,

$$M \leq 2\left\lfloor \frac{9}{18-16} \right\rfloor = 2\left\lfloor \frac{9}{2} \right\rfloor = 8.$$

Hence, with $n = 16$ and $d = 9$ we can say that there is a $(16,M,9)$ code where $M \leq 8$. One such code (a trivial one with $M = 2$) is the code:

$$\begin{bmatrix} 1111111111111111 \\ 0000000001111111 \end{bmatrix}.$$

The best $(16,M,9)$ code is a code with $M = 6$ [2, p.674 or appendix A].

Thus, for a given n and d, where $n < 2d$, an (n,M,d) code exists where M is at least 2 and, of course, is bounded above by the result proven in theorem 1.2. On the other hand, if $n \geq 2d$, which is often the case, no general result is known. However, in appendix A many special cases are cited.

We next state the fact that for any two binary n-tuples x and y,

$$\mathrm{wt}(x+y) = \mathrm{wt}(x) + \mathrm{wt}(y) - 2\mathrm{wt}(x * y), \tag{1.5}$$

where $\mathrm{wt}(x)$ represents the weight of the binary n-tuple x and where $x * y$ is the binary n-tuple formed by component-wise binary multiplication of x and y. Thus $\mathrm{wt}(x * y)$ equals the number of 1's that x and y have in common. The truth of this equation is easily established by noting that for every pair of coordinates that x and y have in common, $\mathrm{wt}(x) + \mathrm{wt}(y)$ must be reduced by 2 in order to maintain equality with $\mathrm{wt}(x + y)$.

In the following chapters we will examine methods of nonlinear code construction and will typically want to know the values of n, M, and d. In the remaining part of this chapter some additional theorems are developed to aid us in that endeavor. It will be convenient to use the notation $A(n,d)$ to represent the *largest number, M, of codewords* in an (n,M,d) code. The next theorem shows that it is sufficient to know $A(n,d)$ where d is even (or odd).

Theorem 1.3 [1]

$$A(n,2r-1) = A(n+1,2r).$$

proof:

We will show that corresponding to any $(n,M,2r-1)$ code C, there is an $(n+1,M,2r)$ code C', and conversely, from which it follows that the maximum number of codewords in C and C' is the same. Extend C by adding an overall parity check. That is, put a 0 at the end of every codeword of even weight and a 1 at the end of every codeword of odd weight. Denote the code thus created by C' and observe that by equation (1.5), C' is an $(n+1,M,2r)$ code. Now puncture C' by deleting one coordinate from each codeword. If the deleted coordinate (from all codewords) is the same, then the resulting code has minimum distance $2r$, otherwise the minimum distance is $2r - 1$. Thus, the code that results, after deletion of a coordinate from each codeword of C' is an $(n,M,2r-1)$ code. Therefore, $A(n,2r-1) = A(n+1,2r)$.

Example 1.7

In chapter three, nonlinear codes known as the Nordstrom-Robinson codes are studied. One of these is a (15,256,5) code and it is known that 256 is the maximum number of possible codewords. Now by theorem 1.3 we can calculate that $A(16,6) = A(15,5) = 256$ also. This says that there is a (16,256,6) code where again 256 is the maximum number of codewords. This latter code is also a Nordstrom-Robinson code.

The next theorem is useful for determining a bound on $A(n,d)$ when the value of $A(n-1,d)$ is known.

Theorem 1.4 [3]

$$A(n,d) \leq 2A(n-1,d).$$

proof:

Consider a code with $A(n,d)$ codewords of length n and minimum distance d. Separate these codewords into two classes, namely those that begin with a 0 and those that begin with a 1. Note that at least one of the two classes will contain at least one-half of the codewords. Now delete the first component of each codeword in that class. The collection of codewords that remains forms an $(n-1,M,d)$ code where $M \geq \frac{A(n,d)}{2}$. And thus, $A(n,d) \leq 2M \leq 2A(n-1,d)$.

In the following corollary some more results regarding bounds on the maximum number of codewords in a nonlinear code are stated. These results can be established as consequences of theorems 1.2, 1.3, and 1.4.

Corollary 1.5 [3]

If d is even then

a) $A(n,d) \leq 2\left\lfloor \frac{d}{2d-n} \right\rfloor$, if $2d > n$,

b) $A(2d,d) \leq 4d$.

If d is odd then

c) $A(n,d) \leq 2\left\lfloor \frac{d+1}{2d+1-n} \right\rfloor$, if $2d+1 > n$,

d) $A(2d+1,d) \leq 4d+4$.

We are now ready to turn our attention to the study of various nonlinear error-correcting codes.

BIBLIOGRAPHY

[1] R.W. Hamming, Error Detecting and Error Correcting Codes, *Bell System Technical Journal*, **29** (1950), 147-160.

[2] F.J. MacWilliams and N.J.A. Sloane, *The Theory of Error-Correcting Codes*, North-Holland Amsterdam, 1977.

[3] M. Plotkin, Binary Codes with Specified Minimum Distance, *IEEE Transactions on Information Theory*, **6** (1960), 445-450.

CHAPTER 2

Hadamard Matrices and Hadamard Codes

INTRODUCTION

The first work with Hadamard matrices was done by J. J. Sylvester in 1867 [10]. Later in 1893, J. Hadamard found a result concerning the maximum value of a determinant. The associated matrices were subsequently named in honor of Hadamard. There are applications of Hadamard matrices in error-correction, information transfer, and combinatorial configurations. In this chapter three different constructions for Hadamard codes are presented.

PROPERTIES OF HADAMARD MATRICES

We begin with a basic definition.

Definition 2.1

A *Hadamard matrix*, H, of order n, is an $n \times n$ matrix whose elements are +1's and −1's such that $HH^T = nI$.

Thus, for a Hadamard matrix the traditional dot product of any two distinct rows is 0 (i.e., any two distinct rows are orthogonal), and the traditional dot product of any row with itself is n.

It is easy to see from definition 2.1 that $HH^T = nI$ implies $H^{-1} = \frac{1}{n}(H^T)$. Thus,

$$\frac{1}{n}H^T H = I,$$

and hence,

$$H^T H = nI.$$

Consequently, the columns of a Hadamard matrix have the same properties as the rows.

Theorem 2.1 [4, p.44]

If any row (or column) of a Hadamard matrix is multiplied by -1 the result is another Hadamard matrix.

proof:

It will suffice to prove this theorem for the case of rows. Assume that some row has been multiplied by -1. Denote that result as row r. Clearly, the dot product of any row (including r) with itself is still n. Next, consider the dot product of two distinct rows. Obviously, here we only need to concern ourselves with row r and any other row r'. When forming the dot product of r and r', products that were 1 will now be -1 and vice versa. Thus, $r \cdot r'$ is equal to zero.

Definition 2.2

A *normalized Hadamard matrix*, H, is a Hadamard matrix where the first row and the first column are all $+1$'s.

Example 2.1

There is a normalized Hadamard matrix of order one, namely, $\begin{bmatrix} 1 \end{bmatrix}$. There is a normalized Hadamard matrix of order two which is

$$\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

There is also a normalized Hadamard matrix of order four. It is

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}.$$

That each of the three illustrations in this example is a Hadamard matrix can be verified by checking to see that $HH^T = nI$.

It is easy to see by theorem 2.1 that any Hadamard matrix can be normalized. Hence, without loss of generality, we do so in the next theorem which specifies a necessary condition for the existence of a Hadamard matrix.

Theorem 2.2 [4, p.44]

If a Hadamard matrix of order n exists, denoted by H_n, then $n = 1$, 2, or a multiple of 4.

proof:

By example 2.1 we know that Hadamard matrices of order one and two exist. Thus, we assume $n \geq 3$ and that H_n is normalized. Also, we can permute rows as well as columns, so we depict the first three rows as:

row 1:	11...1	11...1	11...1	11...1
row 2:	11...1	11...1	--...-	--...-
row 3:	11...1	--...-	11...1	--...-
	i	j	k	m

where for brevity and neatness, we use - to represent -1 and i, j, k, and m represent the number of bits in the successive groups. We know that distinct rows are orthogonal which implies

$$i + j - k - m = 0 \text{ (from orthogonality of rows 1 \& 2)},$$
$$i - j + k - m = 0 \text{ (from orthogonality of rows 1 \& 3)},$$
$$i - j - k + m = 0 \text{ (from orthogonality of rows 2 \& 3)}.$$

These equations along with the fact that $i + j + k + m = n$ imply that $n = 4i$ and thus n is a multiple of four.

It is, at present, an open question as to whether $n = 1$, 2, or a multiple 4, is a sufficient condition for the existence of a Hadamard matrix. Until 1986, the smallest order for which a Hadamard matrix had not been constructed was 268. However, in a recent paper by K. Sawade [8] such a matrix is constructed. We will say more about the construction later in this chapter.

CONSTRUCTION OF HADAMARD MATRICES

We can now show the first of the two constructions for Hadamard matrices of order n that will be presented in this chapter. This construction shows that Hadamard matrices whose orders are powers of two always exist. These matrices are called *Sylvester matrices* after their discoverer J.J. Sylvester [10].

Theorem 2.3 [4, p.45]

If H_n is a Hadamard matrix of order n, then

$$H_{2n} = \begin{bmatrix} H_n & H_n \\ H_n & -H_n \end{bmatrix}$$

is a Hadamard matrix of order $2n$.

proof:

Clearly H_{2n} is $2n \times 2n$ and has +1's and −1's as its elements. It remains to show then that $H_{2n}H_{2n}^T = 2nI$. We can write

$$H_{2n}H_{2n}^T = \begin{bmatrix} H_n & H_n \\ H_n & -H_n \end{bmatrix} \begin{bmatrix} H_n^T & H_n^T \\ H_n^T & -H_n^T \end{bmatrix}$$
$$= \begin{bmatrix} H_nH_n^T + H_nH_n^T & H_nH_n^T - H_nH_n^T \\ H_nH_n^T - H_nH_n^T & H_nH_n^T + H_nH_n^T \end{bmatrix}$$
$$= \begin{bmatrix} nI + nI & 0 \\ 0 & nI + nI \end{bmatrix}$$
$$= \begin{bmatrix} 2nI & 0 \\ 0 & 2nI \end{bmatrix} = 2nI.$$

Example 2.2

From example 2.1 we saw that

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

Thus applying theorem 2.3 we obtain

$$H_4 = \begin{bmatrix} H_2 & H_2 \\ H_2 & -H_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}.$$

By successive applications of theorem 2.3 we can thus obtain any normalized Hadamard matrix of order 2^n .

In order to show the second intended construction of a Hadamard matrix we need some additional tools, one of which is the meaning and behavior of quadratic residues modulo p.

Definition 2.3

Let p be an odd prime. The nonzero squares modulo p, that is, the numbers 1^2, 2^2, 3^2,... modulo p are called the *quadratic residues* modulo p.

Next we state, without proof, some properties of quadratic residues modulo p that can be found in any number theory book. See, for example, Niven and Zuckerman [5].

1) To find all quadratic residues modulo p it is sufficient to consider the squares: $1^2, 2^2, 3^2, \ldots, \frac{1}{2}(p-1)^2 \bmod p$.
2) The quadratic residues $1^2, 2^2, 3^2, \ldots, \frac{1}{2}(p-1)^2$ are all distinct.

   As a consequence, note that this implies that there exist exactly $\frac{1}{2}(p-1)$ quadratic residues and necessarily, the same number of non-quadratic residues modulo p.
3) The product of two quadratic residues is a quadratic residue.
4) The product of two non-quadratic residues is a quadratic residue.
5) The product of a non-quadratic residue with a quadratic residue is a non-quadratic residue.
6) If p has the form $4k+1$, then -1 is a quadratic residue modulo p.
7) If p has the form $4k+3$, then -1 is a non-quadratic residue modulo p.

Another tool that we need to understand before presenting a second construction of a Hadamard matrix is the Legendre symbol which is now defined.

Definition 2.4

The *Legendre symbol* $\left(\frac{i}{p}\right)$ where i is an integer is defined by the following:

a) $\left(\frac{i}{p}\right) = 0$, if i is a multiple of p,

b) $\left(\frac{i}{p}\right) = 1$, if i is a quadratic residue modulo p,

c) $\left(\frac{i}{p}\right) = -1$, if i is a non-quadratic residue modulo p.

Now we state, without proof, two properties of the Legendre symbol. [5]

1) If $0 \leq x, y \leq p-1$, then $\left(\frac{xy}{p}\right) = \left(\frac{x}{p}\right)\left(\frac{y}{p}\right)$.

2) If $c \not\equiv 0 \bmod p$, then $\sum_{b=0}^{p-1}\left(\frac{b}{p}\right)\left(\frac{b+c}{p}\right) = -1$.

We next use the Legendre symbol to define a Jacobsthal matrix which will enable us to show a second construction of Hadamard matrices.

Definition 2.5

A *Jacobsthal matrix* $Q = (q_{ij})$ is a $p \times p$ matrix whose rows and columns are indexed by $0, 1, 2, \ldots, p-1$, where p is a prime power of the form $4k+3$, and $q_{ij} = \left(\frac{j-i}{p}\right)$.

As a result of the definition above we can see that Jacobsthal matrices have dimensions 3×3, 7×7, 11×11, 19×19, etc.

Example 2.3

The 3×3 Jacobsthal matrix is

$$\begin{bmatrix} 0 & 1 & -1 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{bmatrix}.$$

Theorem 2.4 [4, p.47]

A Jacobsthal matrix Q is skew-symmetric, that is, $Q^T = -Q$.

proof:

In order to show that $Q^T = -Q$ we must show that $q_{ij} = -q_{ji}$. We can write:

$$q_{ij} = \left(\frac{j-i}{p}\right) = \left(\frac{-1}{p}\right)\left(\frac{i-j}{p}\right) = \left(\frac{-1}{p}\right)q_{ji}.$$

However, since p is of the form $4k + 3$, we know that -1 is a non-quadratic residue which implies that $\left(\frac{-1}{p}\right) = -1$. Hence, $q_{ij} = -q_{ji}$.

It might have been noticed from definition 2.5 that there are always 0's on the main diagonal of a Jacobsthal matrix. It is also true that each row of a $p \times p$ Jacobsthal matrix has $\frac{1}{2}(p-1)$ +1's and $\frac{1}{2}(p-1)$ −1's. This follows since any row contains the elements $\left(\frac{0}{p}\right), \left(\frac{1}{p}\right), \ldots, \left(\frac{p-1}{p}\right)$ (in some order) and from the fact that in $\mathbb{Z}_p$ (p an odd prime) there are $\frac{1}{2}(p-1)$ quadratic residues and $\frac{1}{2}(p-1)$ non-quadratic residues. Moreover, the same result is true for columns of a Jacobsthal matrix since $Q^T = -Q$.

We will need the following theorem.

Theorem 2.5 [4, p.47]

For any $p \times p$ Jacobsthal matrix Q, if J is a $p \times p$ square matrix of all 1's and O is a $p \times p$ matrix of all 0's, then

$$QQ^T = pI - J \text{ and}$$
$$QJ = JQ = \mathrm{O}.$$

proof:

Let us first prove that $QQ^T = pI - J$. It is easily seen that

$$(QQ^T)[i,i] = \sum_{k=0}^{p-1} Q[i,k]^2 = p - 1,$$

since the number of +1's and the number of −1's in any row of Q is $\frac{1}{2}(p-1)$. Next we observe that if $i \neq j$, then

$$(QQ^T)[i,j] = \sum_{k=0}^{p-1} Q[i,k]Q[j,k] = \sum_{k=0}^{p-1} \left(\frac{k-i}{p}\right)\left(\frac{k-j}{p}\right).$$

Now let $b = k - i$ and $c = i - j \neq 0$. Thus, $b + c = k - i + i - j = k - j$, and so

$$(QQ^T)[i,j] = \sum_{b=-i}^{p-1-i} \left(\frac{b}{p}\right)\left(\frac{b+c}{p}\right).$$

But, summing from $b = -i$ to $b = p - 1 - i$ will produce the same result as when summing from $b = 0$ to $b = p - 1$. Thus $(QQ^T)[i,j] = -1$. So, $QQ^T = pI - J$. The remaining part of the proof follows immediately by again using the fact that the number of +1's and the number of −1's in any row (or column) is $\frac{1}{2}(p-1)$. This implies that $QJ = JQ = O$.

We now can finally show a second construction of Hadamard matrices. We do so with another theorem.

Theorem 2.6 [6]

If Q is a $p \times p$ Jacobsthal matrix and 1_p is a $1 \times p$ row vector of all 1's then

$$H = \begin{bmatrix} 1 & 1_p \\ 1_p^T & Q - I \end{bmatrix}$$

is a Hadamard matrix of order $p + 1$.

proof:

It is obvious that H, as defined above, is a square matrix with dimensions $(p+1) \times (p+1)$. It remains for us to show that $HH^T = (p+1)I$. Observe,

$$HH^T = \begin{bmatrix} 1 & 1_p \\ 1_p^T & Q - I \end{bmatrix} \begin{bmatrix} 1 & 1_p \\ 1_p^T & Q^T - I \end{bmatrix}$$

$$= \begin{bmatrix} (p+1) & O \\ O & J + (Q - I)(Q^T - I) \end{bmatrix},$$

and

$$J + (Q - I)(Q^T - I) = J + QQ^T - QI - IQ^T + I^2.$$

Whence, by theorem 2.4 and theorem 2.5,

$$J + (Q - I)(Q^T - I) = J + pI - J - Q + Q + I = (p + 1)I.$$

Consequently, we have shown that H is a Hadamard matrix of order $p + 1$, as claimed.

This is easily generalized to $p^\alpha + 1$, where p^α is prime power [2, p. 91].

The constructed normalized Hadamard matrices in theorem 2.6 are said to be of *Paley type.*

Example 2.4

Using the Jacobsthal matrix constructed in example 2.3, a 4×4 Hadamard matrix can be constructed by appending a top row and far left column of four 1's and by subtracting 1 from each of the diagonal elements. We get

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & -1 \end{bmatrix}.$$

This is seen to be the same as the Hadamard matrix H_4 in example 2.2 once the third and fourth rows are permuted.

Note, of course, that the first construction for Hadamard matrices that we considered produces a matrix whose order is a power of two, whereas the second construction produces a Hadamard matrix of order $p^\alpha + 1$ which is a multiple of four. We can now show how to form the *Hadamard codes.*

BINARY HADAMARD MATRICES AND HADAMARD CODES

We begin this section with a definition that leads directly to the Hadamard codes.

Definition 2.6

Let H_n be a normalized Hadamard matrix of order n. Map +1's to 0's and −1's to +1's. The resulting matrix is called the *binary Hadamard matrix* $\mathbf{A}_n$.

Example 2.5

Using the matrix H_4 from example 2.2 we construct the binary Hadamard matrix $\mathbf{A}_4$ shown below.

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

Recall that distinct rows of H_n are orthogonal. Hence, half of the time corresponding elements of distinct rows are the same and half of the time they are different. Thus, once H_n is transformed into $\mathbf{A}_n$, distinct rows of $\mathbf{A}_n$ will differ in $\frac{1}{2}n$ places. As a result, the Hamming distance between any two distinct rows of $\mathbf{A}_n$ is $\frac{1}{2}n$. Moreover, the rows of $\mathbf{A}_n$ never differ in the first coordinate position since that will always be zero. We thus have the following theorem.

Theorem 2.7 [1]

If $\mathbf{A}_n$ is a binary Hadamard matrix, then the matrix A_n formed by deleting the first column of $\mathbf{A}_n$ produces an $(n-1, n, \frac{1}{2}n)$ code (a Hadamard code), denoted as A_n.

A simple illustration of such a code is the code A_4 formed by deletion of the first column of the matrix depicted in example 2.5. This gives the not very useful (3,4,2) Hadamard code.

In order to form more codes consider the binary Hadamard matrix $\mathbf{A}_n$. Next form the matrix $\overline{\mathbf{A}}_n$ by complementing all the bits of $\mathbf{A}_n$. These two matrices together give a collection of $2n$ codewords all of length n. Now consider what the minimum distance will be. We already have established that any two distinct rows of $\mathbf{A}_n$ differ in exactly $\frac{1}{2}n$ places. The same must be true of $\overline{\mathbf{A}}_n$. Hence, any row of $\mathbf{A}_n$ will differ in exactly $\frac{1}{2}n$ or n places from any row of $\overline{\mathbf{A}}_n$. As a result we deduce that the minimum Hamming distance for the collection of codewords formed by $\mathbf{A}_n$ and $\overline{\mathbf{A}}_n$ is $\frac{1}{2}n$. Realizing that the first column of $\mathbf{A}_n$ is all 0's and that the first column of $\overline{\mathbf{A}}_n$ is all 1's tells us that if we were to delete the first column of both of these matrices we would have a collection of codewords whose minimum Hamming distance is $\frac{1}{2}n - 1$. We summarize in the theorem below.

Theorem 2.8 [7]

a) The code B_n formed by the collection of codewords of A_n and $\overline{A}_n$ (the complements of A_n) is an $(n-1, 2n, \frac{1}{2}n - 1)$ code (a Hadamard code).

b) The code C_n formed by the collection of codewords of $\mathbf{A}_n$ and $\overline{\mathbf{A}}_n$ (the complements of $\mathbf{A}_n$) is an $(n, 2n, \frac{1}{2}n)$ code (a Hadamard code).

Example 2.6

To construct a Hadamard code that will be triple error-correcting a minimum Hamming distance of seven is needed. Using a Hadamard code, B_n, implies that with $\frac{1}{2}n - 1 = 7$, n must be 16. Thus, we construct B_{16}. Since the order is a power of two we can construct an appropriate Hadamard matrix using the first construction (a Sylvester matrix). Once that is done we form the corresponding binary Hadamard matrix $\mathbf{A}_{16}$ and delete the first column of zeros to get A_{16}. The resulting (15,32,7) code is given by:

$$B_{16} = \begin{bmatrix} A_{16} \\ \overline{A}_{16} \end{bmatrix},$$

where

$$A_{16} = \begin{bmatrix} 000000000000000 \\ 101010101010101 \\ 011001100110011 \\ 110011001100110 \\ 000111100001111 \\ 101101001011010 \\ 011110000111100 \\ 110100101101001 \\ 000000011111111 \\ 101010110101010 \\ 011001111001100 \\ 110011010011001 \\ 000111111110000 \\ 101101010100101 \\ 011110011000011 \\ 110100110010110 \end{bmatrix}.$$

Observe by corollary 1.5 to the Plotkin bound that this code has the largest possible number of codewords for $n = 15$ and $d = 7$. In fact, by the same corollary each of the codes B_n and C_n is optimal. A list of the best codes for a given n and d, where $n \leq 24$, appears in appendix A.

We now turn our attention to two other constructions of codes which will lead us to the famous theorem of Levenshtein.

OTHER CONSTRUCTIONS AND LEVENSHTEIN'S THEOREM

We begin with a simple construction starting with the Hadamard code A_n. Pick all the codewords in A_n that begin with zero and then delete each leading zero. The resulting code, denoted A_n', has codewords of length $n - 2$. The minimum distance is $\frac{1}{2}n$ since there would be no change in that parameter from the $(n - 1, n, \frac{1}{2}n)$ code, A_n. Moreover, the number of codewords in A_n' must be half that of A_n because any column of a binary Hadamard matrix has half zeros and half ones. Thus, we state the following theorem.

Theorem 2.9 [3]

The code A_n' is an $(n-2, \frac{1}{2}n, \frac{1}{2}n)$ code.

Another construction, known as *pasting*, is derived from any two codes C_1 and C_2, where C_1 is an (n_1, M_1, d_1) code and C_2 is an (n_2, M_2, d_2) code. A new code can be formed by "pasting" together a copies of C_1 and b copies of C_2 as shown in figure 2.1 below.

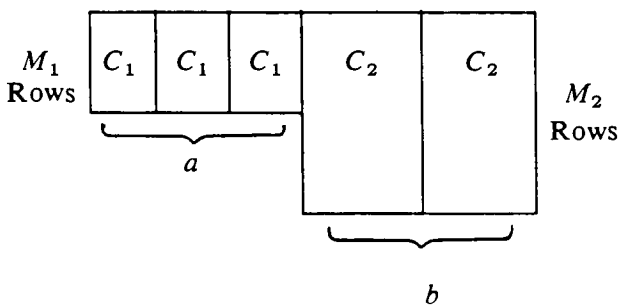


Figure 2.1

In this construction with $M_2 > M_1$ we delete the last $M_2 - M_1$ rows of the larger code, here C_2. Pasting gives codewords of length $n = an_1 + bn_2$. In addition, the number of codewords $M = \min\{M_1, M_2\}$. Also, the minimum distance of the new code is $d = ad_1 + bd_2$. We summarize this in the next theorem.

Theorem 2.10 [3]

Given an (n_1,M_1,d_1) code C_1 and an (n_2,M_2,d_2) code C_2, the code formed by pasting together a copies of C_1 and b copies of C_2, denoted as $aC_1 \oplus bC_2$, is an (n,M,d) code where:

$$n = an_1 + bn_2,$$
$$M = \min\{M_1,M_2\},$$
$$d = ad_1 + bd_2.$$

The result proven in the next theorem is needed in the proof of Levenshtein's theorem.

Lemma 2.11 [3]

If $2d > n \geq d$ and $K = \left\lfloor \frac{d}{2d-n} \right\rfloor$ and a and b are given by the following equations:

$$a = d(2K+1) - n(K+1),$$
$$b = Kn - d(2K-1),$$

then a and b are non-negative integers and

$$n = a(2K-1) + b(2K+1),$$
$$d = aK + b(K+1).$$

proof:

In order to prove that a is non-negative we assume the contrary. Thus, we deduce:

$$n(K+1) > d(2K+1),$$
$$nK + n > 2dK + d,$$
$$n > (2d-n)K + d.$$

We now note that $(2d-n)K$ can assume the value d and when it does the last inequality above becomes $n > 2d$, which is clearly a contradiction. Thus $a \geq 0$. In order to show that b is non-negative let us assume the contrary. Hence,

$$d(2K-1) > Kn,$$
$$(2d-n)K > d,$$
$$K > \frac{d}{2d-n},$$

which again gives a contradiction. Thus $b \geq 0$. To show that n and d are given by the stated equations above is a simple matter of solving the given equations for n and d.

We are now ready to state and prove Levenshtein's theorem.

Theorem 2.12 Levenshtein's Theorem [3]

If enough Hadamard matrices exist, then equality holds true in the Plotkin bound relations (corollary 1.5). That is, if enough Hadamard matrices exist then, if d is even:

a) $A(n,d) = 2\left\lfloor \frac{d}{2d-n} \right\rfloor$, if $2d > n \geq d$,

b) $A(2d,d) = 4d$,

and if d is odd:

c) $A(n,d) = 2\left\lfloor \frac{d+1}{2d+1-n} \right\rfloor$, if $2d+1 > n \geq d$,

d) $A(2d+1,d) = 4d+4$.

proof:

In order to prove (a) we will construct an (n,M,d) code with $M = 2\left\lfloor \frac{d}{2d-n} \right\rfloor$ for any n where d is even and $2d > n \geq d$. Use will be made of lemma 2.11 and we will consider three subcases:

i) n is even,
ii) n is odd and K is even,
iii) n is odd and K is odd.

In case (i) with n even we have a and b of lemma 2.11 both even. In case (ii) with n odd and K even we find a is odd and b is even. In case (iii) with n and K odd we conclude a is even and b is odd. Next, by pasting, we form the code C where:

$$C = \begin{cases} \frac{1}{2}a\,A'_{4K} \oplus \frac{1}{2}b\,A'_{4K+4}, \text{ in case (i);} \\ a\,A_{2K} \oplus \frac{1}{2}b\,A'_{4K+4}, \text{ in case (ii);} \\ \frac{1}{2}a\,A'_{4K} \oplus b\,A_{2K+2}, \text{ in case (iii).} \end{cases}$$

If we can now show that C has length n, minimum distance d and exactly $M = 2\left\lfloor \frac{d}{2d-n} \right\rfloor$ codewords we will be finished with the proof of (a). We do so for case (i), the other two cases being very similar. Consider then

$$\tfrac{1}{2}a\,A'_{4K} \oplus \tfrac{1}{2}b\,A'_{4K+4},$$

and note that by theorem 2.9, A'_{4K} is a $(4K-2, 2K, 2K)$ code and A'_{4K+4} is a $(4K+2, 2K+2, 2K+2)$ code. Hence, by pasting together $\frac{1}{2}a$ copies of A'_{4K} and $\frac{1}{2}b$ copies of A'_{4K+4} we get a code where

$$\begin{aligned} \text{length} &= (\tfrac{1}{2}a)(4K-2) + (\tfrac{1}{2}b)(4K+2) \\ &= a(2K-1) + b(2K+1), \end{aligned}$$

which by lemma 2.11 gives length $= n$. Next we observe that by theorem 2.10 and lemma 2.11, the minimum distance is $(\frac{1}{2}a)(2K) + (\frac{1}{2}b)(2K+2)$, which simplifies to $aK + b(K+1) = d$. Finally, the number of codewords is

$$\min\{2K, 2K+2\} = 2K = 2\left\lfloor \frac{d}{2d-n} \right\rfloor,$$

thus establishing (a).

In order to prove (b) consider the $(n, 2n, \frac{1}{2}n)$ code C_n of theorem 2.8 (b) and let $n = 2d$ where d is even. We thus have a $(2d, 4d, d)$ code and $A(2d,d) = 4d$.

In order to prove (c), knowing that d is odd allows us to use theorem 1.3 and we have $A(n,d) = A(n+1,d+1)$. Now applying part (a) we obtain

$$A(n+1,\, d+1) = 2\left[\frac{d+1}{2(d+1)-(n+1)}\right] = 2\left[\frac{d+1}{2d+1-n}\right],$$

where

$$2(d+1) > n+1 \quad \text{and} \quad 2d+1 > n.$$

which completes the proof of part (c).

Finally, in order to establish part (d), knowing that d is odd we have:

$$A(2d+1,d) = A(2d+2,d+1).$$

And by (b) the right-hand side is equal to $4(d+1) = 4d+4$.

As a result of Levenshtein's theorem we know that there do exist codes which meet the Plotkin bound. Of course the Plotkin bound is deducible for any code for which $2d > n$. Naturally, that is not always the case and in the next chapter we examine such a code.

THE HADAMARD MATRIX H_{268}

Earlier in this chapter it was mentioned that until 1986, the smallest order for which a Hadamard matrix had not been constructed was 268. Now, of course, H_{268} is known and we proceed to briefly discuss how it was discovered by Sawade. First needed are some definitions.

Definition 2.7

A *symmetric matrix* is a square matrix A such that $A = A^T$.

Definition 2.8

A *circulant* $n \times n$ *matrix* is a matrix of the form

$$\begin{bmatrix} a_1 & a_2 & a_3 & \cdots & a_{n-1} & a_n \\ a_n & a_1 & a_2 & \cdots & a_{n-2} & a_{n-1} \\ \cdot & & & & & \cdot \\ \cdot & & & & & \cdot \\ \cdot & & & & & \cdot \\ a_2 & a_3 & a_4 & \cdots & a_n & a_1 \end{bmatrix}.$$

Definition 2.9

The $q \times q$ matrices A, B, C, D whose elements are 0, +1, or −1 are called *T-matrices* provided:

a) A, B, C, D are circulant matrices,

b) $|a_{ij}| + |b_{ij}| + |c_{ij}| + |d_{ij}| = 1$, for all $1 \leq i, j \leq q$,

c) $AA^T + BB^T + CC^T + DD^T = qI_q$, where I_q is the $q \times q$ identity matrix.

Definition 2.10

The symmetric $r \times r$ matrices W_i, $i = 1, 2, 3, 4$ whose elements are +1 or −1 are called *Williamson matrices of order r* provided:

a) $W_i W_j = W_j W_i$ for all $1 \leq i, j \leq 4$,

b) $W_1^2 + W_2^2 + W_3^2 + W_4^2 = 4rI_r$, where I_r is the $r \times r$ identity matrix.

With these definitions as a basis, Sawade next observed that by a theorem due to Turyn [11] a Hadamard matrix of order $4qr$ can be constructed. Next he proved the following theorem.

Theorem 2.13

If there exists a Williamson matrix of order r, then there exists a Hadamard matrix of order $268r$.

In view of this theorem it is sufficient to show the existence of T-matrices of order 67 which was done by Sawade with the aid of a computer.

In [8] Sawade mentions that he learned of the construction of H_{412} by Z. Kiyasu. The significance of this is that now the smallest order for which the existence of a Hadamard matrix is undecided is $n = 428$. That deduction is reached by referring to a list of Hadamard matrices made by I.S. William and N. Wormald who used a computer to list all known Hadamard matrices of orders less than 40,000 [9]. The list shows that at the time of its construction in 1978, Hadamard matrices were not yet known for orders 268, 412, or 428. Hence, the only remaining order less than 40,000 for which the existence of a Hadamard matrix is undecided is 428.

BIBLIOGRAPHY

[1] R.C. Bose and S.S. Shrikhande, A Note on a Result in the Theory of Code Construction, *Information and Control*, **2** (1959), 183-194.

[2] A.W. Geramita and J. Seberry, Orthogonal Designs, *Lecture Notes in Pure and Applied Mathematics*, **N 45,** 1979.

[3] V.I. Levenshtein, The Application of Hadamard Matrices to a Problem in Coding, *Problemy Kibernetiki*, **5** (1961), 123-136. English translation in *Problems in Cybernetics*, **5** (1964), 166-184.

[4] F.J. MacWilliams and N.J.A. Sloane, *The Theory of Error-Correcting Codes*, North-Holland Amsterdam, 1977.

[5] I. Niven and H.S. Zuckerman, *An Introduction to the Theory of Numbers*, 3rd edition, John Wiley & Sons Inc., 1972.

[6] R.E.A.C. Paley, On Orthogonal Matrices, *Journal of Mathematics and Physics*, **12** (1933), 311-320.

[7] M. Plotkin, Binary Codes with Specified Minimum Distance, *IEEE Transactions on Information Theory* **6** (1960), 445-450.

[8] K. Sawade, A Hadamard Matrix of Order 268, *Graphs and Combinatorics*, **2** (1986), 185-187.

[9] J. Seberry, A Computer Listing of Hadamard Matrices, *Lecture Notes in Mathematics*, N 686 275-281.

[10] J.J. Sylvester, Thoughts on Inverse Orthogonal Matrices, Simultaneous Successions, and Tesselated Pavements in Two or More Colors, with Applications to Newton's Rule, Ornamental Tile Work, and the Theory of Numbers, *Philosophy Magazine*, **34** (1867), 461-475.

[11] R.J. Turyn, Hadamard Matrices, Baumert-Hall Units, Four-Symbol Sequences, Pulse Compressions, and Surface Wave Encodings, *Journal of Combinatorial Theory (A)* 16 (1974), 313-333.

CHAPTER 3

The Nordstrom-Robinson Codes

INTRODUCTION

In 1966, Professor John P. Robinson of the Department of Electrical Engineering at the University of Iowa gave an introductory talk on coding theory to high school students in East Moline, Illinois. At that time it was known that the best upper bound for the number of codewords in a $(15, M, 5)$ nonlinear code was 2^8 [2], but the code was unknown. Professor Robinson posed the construction of such a nonlinear code to the students. One of the students, Alan W. Nordstrom, found the code. This code is now known as the Nordstrom-Robinson $(15, 256, 5)$ nonlinear code and is listed in appendix A. The code is interesting because it has been shown by Wagner [6] that the best linear code with $n = 15$ and $d = 5$ has only 128 codewords. In this chapter we will show how to construct the $(15, 256, 5)$ Nordstrom-Robinson code, denoted N'_{16} as well as N_{16}, the Nordstrom-Robinson $(16, 256, 6)$ code. However, in order to make some observations in this and future chapters we first give a brief introduction to groups and cosets.

GROUPS AND COSETS

We begin with a basic definition.

Definition 3.1

$[G, *]$ is a *group* provided that G is a nonempty set, $*$ is a binary operation on G, and

a) G is closed under $*$,

b) G is associative under $*$,

c) there is an identity element $i \in G$, that is, $x * i = i * x = x$, for all $x \in G$,

d) each $x \in G$ has an inverse $x^{-1} \in G$, that is, $x * x^{-1} = x^{-1} * x = i$.

A group $[G, *]$ which is commutative under $*$ is called a *commutative* or *abelian* group.

Example 3.1

Let $A = \{1, 2, 3\}$ and consider all bijections which map A onto itself. Using cycle notation:

$$f_1 = (1) = i \qquad f_4 = (2\ 3)$$

$$f_2 = (1\ 2\ 3) \qquad f_5 = (1\ 2)$$

$$f_3 = (1\ 3\ 2) \qquad f_6 = (1\ 3)$$

If we let $S_3 = \{f_1, f_2, f_3, f_4, f_5, f_6\}$ and if we define our binary operation to be function composition 'o' we have:

o	f_1	f_2	f_3	f_4	f_5	f_6
f_1	f_1	f_2	f_3	f_4	f_5	f_6
f_2	f_2	f_3	f_1	f_6	f_4	f_5
f_3	f_3	f_1	f_2	f_5	f_6	f_4
f_4	f_4	f_5	f_6	f_1	f_2	f_3
f_5	f_5	f_6	f_4	f_3	f_1	f_2
f_6	f_6	f_4	f_5	f_2	f_3	f_1

As can be noted by the table above, $[S_3, o]$ is a non-abelian group. Moreover, a set such as S_3 which consists of all bijections of $\{1, 2, 3\}$ onto itself is called a *group of permutations.*

We next define the notion of subgroup which will aid us in understanding the meaning of cosets.

Definition 3.2

Let $[G, *]$ be a group and $A \subseteq G$. Then $[A, *]$ is a *subgroup* of $[G, *]$ written $[A, *] \leq [G, *]$, provided that $[A, *]$ is itself a group.

When it is clear from the context we often use the symbol G to represent the group $[G, *]$ and the symbol A to represent the subgroup $[A, *]$. Next we build on this definition with the following.

Definition 3.3

Let G be a group and $H \leq G$. For any $a \in G$, the set $aH = \{ah : h \in H\}$ is called the *left coset* of H in G containing a. Similarly, $Ha = \{ha : h \in H\}$ is called the *right coset* of H in G containing a.

Example 3.2

From the group in example 3.1 we obtain a subgroup $H = \{f_1, f_6\} = \{(1), (1\ 3)\}$. We display the left cosets below.

Element a	Left Coset of H containing a
(1)	$(1)H = \{(1), (1\ 3)\} = H$
(1 2 3)	$(1\ 2\ 3)H = \{(1\ 2\ 3), (1\ 2)\} = \{f_2, f_5\}$
(1 3 2)	$(1\ 3\ 2)H = \{(1\ 3\ 2), (2\ 3)\} = \{f_3, f_4\}$
(2 3)	$(2\ 3)H = \{(2\ 3), (1\ 3\ 2)\} = \{f_4, f_3\}$
(1 2)	$(1\ 2)H = \{(1\ 2), (1\ 2\ 3)\} = \{f_5, f_2\}$
(1 3)	$(1\ 3)H = \{(1\ 3), (1)\} = H$

In this example, H is a subset of G, a group of permutations. In such case we then refer to H as a *permutation group.*

From the preceding example we note that when $a \in G$ and also $a \in H$, the left coset generated is H itself, whereas if $a \in G$, but $a \notin H$, then the coset generated is not equal to H. Some other facts about cosets are:

1) Two left or right cosets are either identical or disjoint. Thus, a coset is uniquely determined by any *one* of its elements.

2) $aH = bH$ if and only if $a^{-1}b \in H$, which is easy to see, for if we operate on both sides of $aH = bH$ with a^{-1} we get $H = a^{-1}bH$. And this latter result is true if and only if $a^{-1}b \in H$. (Note: $aH = bH$ *does not imply* that $a = b$.)

3) $|aH| = |bH|$, that is, any two cosets of a subgroup H have the same cardinality.

4) $aH = Ha$ if and only if $H = a^{-1}Ha$. (Note: in general $aH \neq Ha$.)

The above facts give the following well-known theorem due to and named after Lagrange.

Theorem 3.1 [Lagrange's Theorem]

If G is a finite group and H is a subgroup of G, then $|G|$ is divisible by $|H|$ and, moreover, the result of this division is equal to the number of cosets.

Applying this theorem to example 3.2 we see $|G| = 6$, $|H| = 2$ and indeed $|G| \div |H|$ is 3 which is the number of distinct left cosets of H in G.

Cosets are intimately related to error-correcting codes as will be seen when we construct N_{16}. First, however, we give the construction of N'_{16} as done by Nordstrom and Robinson [4].

CONSTRUCTION OF N'_{16}

The (15, 256, 5) Nordstrom-Robinson code can be constructed in a fashion similar to that used with parity check matrices and linear codes. We first observe that the rate $R = (\log_2 M)/n = 8/15$. Hence, there are eight message bits denoted $x_0, x_1, \ldots, x_7$, and seven check bits denoted $y_0, y_1, \ldots, y_6$. Also, since there are eight message bits and 256 codewords, it must be the case that all the 256 "values" 00000000, 00000001, 00000010, 00000011, . . . , 11111111 appear in the message bit positions. The check bits are defined by certain nonlinear equations, the first for y_0 being:

$$\begin{aligned} y_0 = {} & (x_0 + x_1 + x_3 + x_6 + x_7) \\ & + (x_0 + x_4)(x_1 + x_2 + x_3 + x_5) \\ & + (x_1 + x_2)(x_3 + x_5), \end{aligned} \tag{3.1}$$

where the additions and multiplications are both modulo 2. The subsequent y's are found by cyclically shifting x_0 through x_6; that is, to determine y_j, we substitute $x_{i+j \bmod 7}$ for x_i in the preceding equation.

The code produced follows in figure 3.1.

x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	y_0	y_1	y_2	y_3	y_4	y_5	y_6
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	1	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	0	0	1	0	1	1
0	0	0	0	0	0	1	1	0	1	1	0	0	1	1
.							.							.
.							.							.
.							.							.
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure 3.1

Preparata [5] has shown that this code has a minimum distance of 5 which will follow as a consequence of the next chapter so for the moment it will be postponed. One other matter worthy of note is that the code N_{16}' can be encoded by using equations (3.1), that is, in hardware with a nonlinear feedback register. Moreover, decoding is also possible as shown in [5].

CONSTRUCTION OF N_{16}

Now we show a construction of the code N_{16} as given by MacWilliams and Sloane [3, *p.* 73]. In order to do so we begin with the well-known Golay code G_{24}. For the reader familiar with linear error-correcting codes we mention that G_{24} is a linear code over $\mathbb{Z}_2$, the integers modulo 2. Moreover, a generator matrix, G, is shown below in figure 3.2.

1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 1 1 0 0 0 1 0
1 0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 1 1 1 0 0 0 1
1 0 0 1 0 0 0 0 0 0 0 0 0 1 0 1 1 0 1 1 1 0 0 0
1 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 1 1 0 1 1 1 0 0
1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 1 1 0 1 1 1 0
1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 1 1 0 1 1 1
1 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 1 0 1 1 0 1 1
1 0 0 0 0 0 0 0 1 0 0 0 0 1 1 0 0 0 1 0 1 1 0 1
1 0 0 0 0 0 0 0 0 1 0 0 0 1 1 1 0 0 0 1 0 1 1 0
1 0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 1 0 0 0 1 0 1 1
1 0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 1 1 0 0 0 1 0 1
0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1

Figure 3.2

Hence, G generates $2^{12} = 4096$ codewords. It is not hard to prove that the minimum distance of G_{24} is 8, and thus in the notation of linear codes, G generates a $[24, 12, 8]$ code.

Next we rewrite G by rearranging the columns to contain the codeword 1111 1111 00...0 = $1^8 0^{16}$ giving the generator matrix G' displayed in figure 3.3.

1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 0 1 0 1 1 0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1
1 0 1 0 1 0 1 0 0 1 0 0 0 0 0 0 0 0 0 1 1 1 0 0
1 0 0 1 1 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 1 0
1 0 0 0 0 1 1 1 0 0 0 1 0 0 0 0 0 0 0 1 0 1 1 0
1 0 0 0 1 0 1 1 0 0 0 0 1 0 0 0 0 0 0 0 1 0 1 1
1 0 1 0 0 1 0 1 0 0 0 0 0 1 0 0 0 0 0 0 1 1 0 1
1 0 1 1 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 1
1 0 1 1 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 1 1 0 1 0
1 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 1 0 0 1 0 1 0 1
1 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 1 1
0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1

Figure 3.3

Now note that the first seven columns of G' are linearly independent while the first eight columns are linearly dependent. Moreover, since the octuples in the first eight positions of each row have even weight, it follows that when G' generates G_{24}, the first seven coordinates may be considered information symbols and the eighth coordinate is the modulo 2 sum of the first seven. Additionally, we can list the codewords of G_{24} whose first seven coordinates are the same, one after the other. With 2^{12} codewords in G_{24} and 2^7 permutations of the first seven bits we obtain $2^{12} / 2^7 = 32$ codewords per group. The result is displayed in figure 3.4 where we have first listed all codewords beginning with eight zeros, followed by those that have a 1 in the eighth position and one other 1 in exactly one of the first seven positions.

0000000 0	32 sixteen bit vectors
1000000 1	32 sixteen bit vectors
0100000 1	32 sixteen bit vectors
. . .	. . .
0000001 1	32 sixteen bit vectors
others	. . .

Figure 3.4

Although $|G_{24}| = 4096$ we have only partially displayed 256 rows, the reason for which is made clear in the next definition.

Definition 3.4

The *Nordstrom-Robinson code* N_{16} is obtained from the first 256 rows of figure 3.4 by deleting the first eight components of each vector. (See remaining boxes.)

Thus, in particular, N_{16} has $n = 16$ and $M = 256$. Moreover, since the Golay code G_{24} has a minimum distance of 8 and over the first eight coordinates any pair of the first 256 codewords differ in at most two positions (see figure 3.4), it follows that N_{16} has a minimum distance of 6.

Next we wish to show that N_{16} is not linear. Denote by abx the concatenation of vectors a, b and x, where a is a seven component vector, b is a one component vector and x is a 16 component vector. Thus, abx can be used to represent a codeword of G_{24} and after deletion of the first eight coordinates (i.e., ab) we are left with x, a codeword of N_{16}.

Now let x_1 be a codeword in N_{16} where $a_1b_1 = 10000001$. Also, let x_2 be a codeword in N_{16} where $a_2b_2 = 01000001$. Hence, $a_1b_1x_1 + a_2b_2x_2 = (11000000)(x_1 + x_2) \in G_{24}$, since G_{24} is a linear code. We now claim that $(x_1 + x_2)$ is not a codeword in N_{16}. In order to verify this claim, suppose the contrary, that is, $x_1 + x_2 \in N_{16}$. Let the eight bits that were deleted to form $x_1 + x_2$ be a_3b_3, where clearly a_3b_3 contains either zero or two 1's (see figure 3.4). But then $(11000000)(x_1 + x_2) + a_3b_3(x_1 + x_2)$ would not have weight of at least eight (a property of codewords in G_{24}). Hence, $x_1 + x_2 \notin N_{16}$. We have now proved the theorem which follows.

Theorem 3.2

The Nordstrom-Robinson code N_{16} is a (16, 256, 6) code which is not linear.

As a final matter about N_{16} we redisplay it below (figure 3.5) again embedded in G_{24}, but this time with some additional notation.

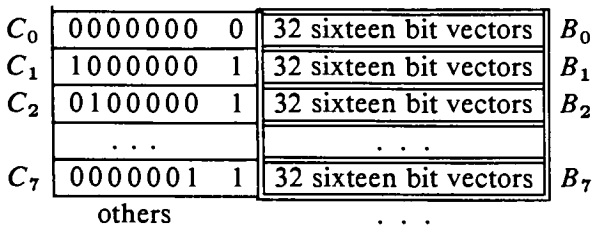


Figure 3.5

The 256 codewords of N_{16} are denoted by the collections B_i, where $i = 0, 1, 2, \ldots, 7$. (See double boxes.) The collection, C_0, of 24 bit codewords includes all codewords in G_{24} that begin with eight zeros (including 0^{24}). Hence, C_0 is closed under addition modulo 2 and consequently $C_0 \leq G_{24}$. Deleting the leading eight zeros of the codewords in C_0 yields B_0 which is thus seen to be a [16, 5, 8] linear code. Moreover, by Lagrange's theorem, G_{24} is partitioned into 128 cosets of C_0 and as a result $B_1, \ldots, B_7$ are seen to be cosets of B_0. This observation will be useful in the next chapter where the Preparata codes which are a generalization of the Nordstrom-Robinson codes are considered.

One final comment is that if one digit is dropped from the (16, 256, 6) Nordstrom-Robinson code N_{16}, a code which is equivalent to the (15, 256, 5) Nordstrom-Robinson code N'_{16} is obtained [1], and as mentioned earlier, $A(15,5) = A(16,6) = 256$ which is twice as many codewords as can be found in the best linear code with $n = 15$ and $d = 5$.

BIBLIOGRAPHY

[1] J.M. Goethals, On the Golay Perfect Binary Code, *Journal of Combinatorial Theory*, **11** (1971), 178-186.

[2] S.M. Johnson, A New Upper Bound for Error-Correcting Codes, *IEEE Transactions on Information Theory*, **8** (1962), 203-207.

[3] F.J. MacWilliams and N.J.A. Sloane, *The Theory of Error-Correcting Codes*, North-Holland Amsterdam, 1977.

[4] A.W. Nordstrom and J.P. Robinson, An Optimum Non-linear Code, *Information and Control*, **11** (1967), 613-616.

[5] F.P. Preparata, A Class of Optimum Non-linear Double-Error-Correcting Codes, *Information and Control*, **13** (1968), 378-400.

[6] T.J. Wagner, A Remark Concerning the Minimum Distance of Binary Group Codes, *IEEE Transactions on Information Theory*, **11** (1965), 458.

CHAPTER 4

The Preparata Codes

INTRODUCTION

The Preparata codes have an interesting history. Preparata [5] discovered this class of codes by studying the properties of the smallest code in the class, the Nordstrom-Robinson (15,256,5) code N'_{16}. The Preparata codes are a generalization of the Nordstrom-Robinson codes and are a class of nonlinear error-correcting codes of length $2^n - 1$ ($n \geq 4$ and even). In addition, they have the largest possible number of codewords for their length and minimum distance.

It is possible to construct these codes as the set union of a particular linear code and of a subset of its cosets. Moreover, the linear code is constructed by using the well-known linear t-error-correcting BCH codes which were discovered in 1959 by the French mathematician A. Hocquenghem [2] and independently in 1960 by R.C. Bose and D.K. Ray-Chauduri [1]. We will summarize the essential facts about BCH codes, but first we give a brief introduction to rings and fields as a needed preliminary.

RINGS AND FIELDS

Definition 4.1

A set R with two binary operations, usually called addition and multiplication, denoted by $[R, +, \cdot]$ is called a *ring* provided

a) $[R, +]$ is an abelian group,

b) $(a \cdot b) \cdot c = a \cdot (b \cdot c)$ for all $a, b, c \in R$,

c) $a \cdot (b + c) = a \cdot b + a \cdot c$ and $(b + c) \cdot a = b \cdot a + c \cdot a$ for all $a, b, c \in R$.

When it is clear from the context, $a \cdot b$ will be symbolized as ab.

A ring $[R, +, \cdot]$ which is commutative with respect to multiplication is called a *commutative ring.*

Example 4.1

a) The set of integers $\{0, 1, \ldots, n-1\}$ under ordinary addition and multiplication modulo n is a commutative ring with multiplicative identity 1.

b) The set $Z[x]$ of all polynomials in the variable x with integer coefficients under ordinary addition and multiplication is a commutative ring with multiplicative identity $f(x) = 1$.

Definition 4.2

Let $[R, +, \cdot]$ be a ring and $S \subseteq R$. Then $[S, +, \cdot]$ is a *subring* of $[R, +, \cdot]$ provided that $[S, +, \cdot]$ is itself a ring.

In any ring, since inverses under addition always exist, we often speak of the operation of subtraction as meaning $a - b = a + (-b)$. With that in mind, the following theorem is easy to prove.

Theorem 4.1

A nonempty subset S of a ring R is a subring of R if

a) $a - b \in S$, and

b) $ab \in S$ for all $a, b \in S$.

Definition 4.3

A subring A of a ring R is called an *ideal* of R if for every $r \in R$ and every $a \in A$, $ra, ar \in A$.

Hence, we say that a subring A of a ring R is an ideal of R if A "absorbs" elements of R, that is, if $rA \subseteq A$ and $Ar \subseteq A$ for all $r \in R$.

Example 4.2

a) For any positive integer n, the set $n\mathbb{Z} = \{0, \pm 1n, \pm 2n, \cdots\}$ is an ideal of $\mathbb{Z}$.

b) Let $\mathbb{R}[x]$ denote the set of all polynomials with real coefficients and A the subset of all polynomials with constant term equal to zero. Then A is an ideal of $\mathbb{R}[x]$.

c) Let R be a commutative ring with a multiplicative identity and let $a \in R$. The set $\langle A \rangle = \{ra : r \in R\}$ is an ideal of R called the *principal ideal generated by a.*

Definition 4.4

A *field* is a ring $[R, +, \cdot]$ for which $[R - \{0\}, \cdot]$ is an abelian group.

Thus, in a field, inverses under multiplication always exist (except for 0). Consequently, we may think of a field as an algebraic system that is closed under addition, subtraction, multiplication, and division (except division by 0) and, moreover, ab^{-1} means a divided by b.

It is finite fields that are of particular interest, however, because they provide a vehicle by which multiple-error-correcting codes can be constructed. The *order of a finite field* is the number of elements in it, and a finite field is usually denoted as GF(n) where n is the order of the field. The letters GF stand for *Galois field* in honor of their discoverer Evariste Galois (1811-1832). Some useful facts about finite fields are:

1) If p is prime, then the integers modulo p form a finite field, denoted by GF(p).
2) Let $f(x)$ be an irreducible polynomial of degree m over GF(p). Then the set of polynomials of degree less than m, with the operations of addition, subtraction, multiplication, and division (modulo $f(x)$), form a finite field F, where F has p^m elements. Moreover, we can view F as the set of polynomials in α, where α is a root of $f(x)$.
3) Every finite field has p^m elements for some prime p and is unique to within an isomorphism. It is denoted GF(p^m).

Example 4.3

From part (3) above, let $p = 2$ and $m = 4$. Then in GF(2) consider the set of polynomials of degree ≤ 3 modulo $x^4 + x + 1$. This gives the finite field GF(2^4) which follows.

Power of α	Polynomial in α	4-tuple
0		(0 0 0 0)
1	1	(1 0 0 0)
α	α	(0 1 0 0)
α^2	α^2	(0 0 1 0)
α^3	α^3	(0 0 0 1)
α^4	$1 + \alpha$	(1 1 0 0)
α^5	$\alpha + \alpha^2$	(0 1 1 0)
α^6	$\alpha^2 + \alpha^3$	(0 0 1 1)
α^7	$1 + \alpha + \alpha^3$	(1 1 0 1)
α^8	$1 + \alpha^2$	(1 0 1 0)
α^9	$\alpha + \alpha^3$	(0 1 0 1)
α^{10}	$1 + \alpha + \alpha^2$	(1 1 1 0)
α^{11}	$\alpha + \alpha^2 + \alpha^3$	(0 1 1 1)
α^{12}	$1 + \alpha + \alpha^2 + \alpha^3$	(1 1 1 1)
α^{13}	$1 + \alpha^2 + \alpha^3$	(1 0 1 1)
α^{14}	$1 + \alpha^3$	(1 0 0 1)

Note that $\alpha^4 + \alpha + 1 = 0$. Moreover, α is a *primitive element* of GF(2^4) which means that every nonzero element of GF(2^4) is a power of α. It does not always happen that a root of an irreducible polynomial is a primitive element. A polynomial that has a primitive element as a root is called a *primitive polynomial* and once an irreducible primitive polynomial of GF(p) of degree m is found, a table such as that in example 4.3 can be constructed. In addition, it is known that every finite field has a primitive element.

CYCLIC CODES AND BCH CODES

With the preceding as basic groundwork, we next define cyclic codes, which include many of the most interesting known codes.

Definition 4.5

An $[n,k,d]$ code C over an alphabet F is called *cyclic* provided that $(c_0,c_1,\ldots,c_{n-1}) \in C$ implies that $(c_{n-1},c_0,c_1,\ldots,c_{n-2}) \in C$.

Now let R be the ring of all polynomials with coefficients in GF(q) and let S be the ideal generated by $x^n - 1$. The cosets of $[S, +]$ in $[R, +]$ are called *residue classes* mod S and form a ring called the *residue class ring* R mod S. This residue class ring R mod S (considered as an additive group) is isomorphic to $\mathbf{R}^n$ where $\mathbf{R}^n$ is the set of all words of length n in GF(q).

Example 4.4

Let R be the ring of all polynomials with coefficients in GF(2) and let S be the ideal generated by $x^3 - 1$ (or equivalently, $x^3 + 1$). Then $S = \{(x^3 + 1)r(x) : r(x) \in R\}$. Next form the cosets of $[S, +]$ in $[R, +]$, that is, form the residue classes modulo S, $\{(x^3 + 1)r(x) + r'(x) : r(x), r'(x) \in R\}$. But there are only eight distinct classes which are given by: $\{(x^3 + 1)r(x) + p(x)\}$ where $r(x) \in R$ and $p(x) \in R$ mod $(x^3 + 1)$. Thus we may take as residue class representatives the eight polynomials: 0, 1, x, $x + 1$, x^2, $x^2 + 1$, $x^2 + x$, $x^2 + x + 1$. This collection of elements, with addition defined as addition modulo 2 and multiplication defined as $(r_1(x) + S)(r_2(x) + S) = r_1(x)r_2(x) + S$, for $r_1(x), r_2(x) \in R$ mod $(x^3 + 1)$ form the residue class ring R mod S. Also, it is easy to see that R mod S is isomorphic to $\mathbf{R}^3$, the set of all codewords of length 3 in GF(2).

Thus we will not distinguish between words of length n and polynomials of degree less than n (mod $x^n - 1$). We note also that multiplication by x in R mod S produces the cyclic shift $(c_0,c_1,\ldots,c_{n-1}) \rightarrow (c_{n-1},c_0,c_1,\ldots,c_{n-2})$, and hence, a cyclic code corresponds to an ideal in R mod S. Moreover, every ideal in R mod S (or equivalently, every cyclic code in $\mathbf{R}^n$) is

generated by a polynomial $g(x)$ which divides $x^n - 1$. We call $g(x)$ the *generator* of the cyclic code and $g(x) = \text{LCM}[f_1(x), \ldots, f_r(x)]$, where the $f_i(x)$ are the minimal polynomials of the zeros of $g(x)$.

BCH codes can be considered as a special case of cyclic codes. A few facts about these codes must be mentioned and then the Preparata codes will be constructed.

Definition 4.6

A cyclic code of length n over GF(q) is called a *BCH code of designed distance* δ if its generator $g(x)$ is the least common multiple of the minimal polynomials of β^l, $\beta^{l+1}, \ldots, \beta^{l+\delta-2}$ for some l, where β is a primitive nth root of unity.

Additionally:

1) A BCH code of designed distance δ has minimum distance $d \geq \delta$.

2) Often $l = 1$ is used in definition 4.6 and we obtain the *narrow-sense* BCH codes. In this case if $q = 2$ then β is a primitive nth root of unity in GF(2^m) and the cyclic code in $\mathbf{R}^n$ which consists of all the codewords that have $\beta, \beta^2, \ldots, \beta^{\delta-1}$ as zeros is the $[n = 2^m - 1, k \geq n - mt, d \geq \delta]$ t-error-correcting BCH code.

3) If in (2) above we also require 1 to be a zero of all the codewords, then the code has a minimum distance of $d + 1$.

Example 4.5

Consider the construction of the 4-error-correcting BCH code of length 31 over GF(2). By [4, p. 196ff],

$$g(x) = (1 + x^2 + x^5)(1 + x^2 + x^3 + x^4 + x^5)$$
$$(1 + x + x^2 + x^4 + x^5)(1 + x + x^2 + x^3 + x^5).$$

From which,

$$g(x) = x^{20} + x^{18} + x^{17} + x^{13} + x^{10} + x^9 + x^7 + x^6 + x^4 + x^2 + 1.$$

Thus the codeword $c = 00000000001011000100111011010101$ along with the ten codewords formed from $xc, x^2c, \ldots, x^{10}c$ can be used as the rows of the generator matrix of the code.

CONSTRUCTION OF THE PREPARATA CODES

Consider the polynomials over GF(2) modulo $(x^{2^{n-1}-1} + 1)$ where $n \geq 4$. Thus, any such polynomial $a(x)$ has degree less than or equal to $2^{n-1} - 2$ and we can represent $a(x)$ as the row

vector $[a_{2^{n-1}-2}, a_{2^{n-1}-3}, \ldots, a_0]$, where $a(x) = \sum_{j=0}^{2^{n-1}-2} a_j x^j$. The following notations will be used:

$W[a(x)]$ = the number of nonzero coefficients of $a(x)$,

$d[a(x), b(x)]$ = the Hamming distance between $a(x)$ and $b(x)$,

and note that $d[a(x), b(x)] = W[a(x) + b(x)]$.

Now consider the single-error-correcting BCH code, $\mathbf{B}_1$, of length $2^{n-1} - 1$ over GF(2). Let $\mathbf{B}_2$ be a double-error-correcting BCH code of length $2^{n-1} - 1$ and distance 6 over GF(2). It is easy to see that β and l in definition 4.6 can be chosen so that $\mathbf{B}_2 \subset \mathbf{B}_1$. Finally, the polynomial $\sum_{j=0}^{2^{n-1}-2} x^j = (x^{2^{n-1}-1} + 1)/(x + 1)$ will be denoted by $u(x)$.

Consider all $(2^n - 1)$ - component vectors in GF(2) of the form

$$v = [m(x), i, m(x) + (m(1) + i)u(x) + s(x)], \tag{4.1}$$

where $m(x) \in \mathbf{B}_1$ and $s(x) \in \mathbf{B}_2$, and $i \in \{0, 1\}$.

The collection of all such vectors v is defined to be the code $\mathbf{C}_n$.

Theorem 4.2 [5]

The code $\mathbf{C}_n$ is a linear code with minimum distance of 6.

We can also observe that since $\mathbf{B}_1$ and $\mathbf{B}_2$ are BCH codes they contribute $2^{n-1} - n$ and $2^{n-1} - 2n$ information bits respectively. Thus the linear code $\mathbf{C}_n$ with length $2^n - 1$ has $2^n - 3n + 1$ information bits and minimum distance 6.

Next we state a series of facts proven by Preparata in [5].

Lemma 4.3

a) Consider the polynomial $\psi(x) = (x^{2^{n-1}-1} + 1)/g(x)$ where $g(x)$ generates the single-error-correcting BCH code of length $2^{n-1} - 1$. Then there exists an s, $0 \leq s \leq 2^{n-1} - 2$ such that $(x^s \psi(x))^2 = x^s \psi(x)$.

b) Let $f(x) = x^s \psi(x)$ and $q(x) \in \{0, 1, x, x^2, \ldots, x^{2^{n-1}-2}\}$. Then the polynomial $q(x) + q(x)f(x) \in \mathbf{B}_1$.

c) Construct vectors of the form

$$u = [q(x), 0, q(x)f(x)]. \tag{4.2}$$

If $n \geq 4$, the sum of any two vectors u_1 , u_2 of the form given by equation (4.2) can be written as

$$u_1 + u_2 = v + q + p,$$

where

$v = [m'(x), 0, m'(x) + m'(1)u(x)]$, $m'(x) \in \mathbf{B}_1$, (Note: v $\in \mathbf{C}_n$)

$q = [q(x), 0, q(x)]$,

$p = [0, 0, m''(x)]$, $m''(x) = (q(x) + q(x)f(x)) + m'(x) + m'(1)u(x) \in \mathbf{B}_1$.

If $q(x) = 0$, then $m'(x) = 0$; if $q(x) \neq 0$, then either $m'(x) = 0$ or $m'(x)$ is a trinomial.

d) For any trinomial $m(x) \in \mathbf{B}_1$, $m(\alpha^3) = \alpha^{3s}(\alpha^h + \alpha^{2h})$ where s and h are integers modulo $2^{n-1} - 1$, $h \neq 0$.

e) For $m''(x) = (q(x) + q(x)f(x)) + m'(x) + m'(1)u(x)$ and any $s(x) \in \mathbf{B}_2$,

$$W[m''(x) + q(x) + s(x)] \geq \begin{cases} 4, \text{ if } n \text{ is even} \\ 2, \text{ if } n \text{ is odd}. \end{cases}$$

Finally we can construct the Preparata codes. Construct $(2^n - 1)$ - component vectors of the form

$$w = v + u, \tag{4.3}$$

where v and u are given by equations (4.1) and (4.2) respectively. Thus, w is the sum of two vectors, one an arbitrary member of the linear code $\mathbf{C}_n$ and the other a vector of the form $[q(x), 0, q(x)f(x)]$. Hence we can write

$$w = [m(x) + q(x), i, m(x) + q(x)f(x) + (m(1) + i)u(x) + s(x)] \tag{4.4}$$

where $m(x)$, $q(x)$, i, and $s(x)$ contribute $2^{n-1} - n$, $n - 1$, 1, and $2^{n-1} - 2n$ information bits respectively.

Definition 4.7

The *Preparata code*, denoted $\mathbf{K}_n$, is the code formed by all codewords of the form (4.4).

As a result the Preparata code $\mathbf{K}_n$ has codewords of length $2^n - 1$ and $2^n - 2n$ information bits.

Theorem 4.4

$\mathbf{K}_n$ is a nonlinear code.

proof:

Assume the contrary which implies that if $w_1, w_2 \in \mathbf{K}_n$, then $w_1 + w_2 \in \mathbf{K}_n$. Let $w_1 = v_1 + u_1$ and $w_2 = v_2 + u_2$ be any two codewords of $\mathbf{K}_n$. Then,

$$w_1 + w_2 = (v_1 + v_2) + (u_1 + u_2).$$

By lemma 4.3 (c), we obtain

$$w_1 + w_2 = (v_1 + v_2) + v + q + p.$$

And by letting $v' = v_1 + v_2 + v$,

$$w_1 + w_2 = v' + q + p. \tag{4.5}$$

By theorem 4.2 and lemma 4.3 (c), v' is an arbitrary member of the linear code $\mathbf{C}_n$. Observe that we can rewrite $q + p$ as

$$\begin{aligned} q + p &= [q(x), 0, q(x) + m''(x)] \\ &= [q(x), 0, q(x)f(x) + m'(x) + m'(1)u(x)] \\ &= [q(x), 0, q(x)f(x)] + [0, 0, m'(x) + m'(1)u(x)] \\ &= u' + [0, 0, m'(x) + m'(1)u(x)]. \end{aligned}$$

Hence equation (4.5) can be written

$$w_1 + w_2 = v' + u' + [0, 0, m'(x) + m'(1)u(x)].$$

By assumption, $w_1 + w_2 \in \mathbf{K}_n$. In addition, $v' + u' \in \mathbf{K}_n$. When $m'(x) \neq 0$, $m'(x) \in \mathbf{B}_2$. As a result, $v' + [0, 0, m'(x) + m'(1)u(x)] \notin \mathbf{C}_n$. All this then implies that $w_1 + w_2 \notin \mathbf{K}_n$ and consequently the assumption that $\mathbf{K}_n$ is linear is false. Whence, $\mathbf{K}_n$ is a nonlinear code.

The following important theorem is proven in [5]

Theorem 4.5

For $n \geq 4$, and even, $\mathbf{K}_n$ is a $(2^n - 1, 2^{2^n - 2n}, 5)$ code.

From equation (4.3) it can be seen that $\mathbf{K}_n$ is formed as the union of cosets of $\mathbf{C}_n$ given by $\mathbf{C}_n + u$, where $u = [q(x), 0, q(x)f(x)]$. That is, $\mathbf{K}_n$ is formed as the union of cosets of $\mathbf{C}_n$ in $\mathbf{K}_n$ containing u. Moreover, since no two elements of $[q(x), 0, q(x)f(x)]$ are in the same coset and $|[q(x), 0, q(x)f(x)]| = 2^{n-1}$, we conclude by LaGrange's theorem that $|\mathbf{K}_n| = 2^{n-1}|\mathbf{C}_n|$.

Example 4.6

We can obtain the (15, 256, 5) Nordstrom-Robinson code from $\mathbf{K}_n$ by letting $n = 4$. Then by equation (4.1), $\mathbf{C}_4$ consists of vectors of the form

$$v = [m(x), i, m(x) + (m(1) + i)u(x) + s(x)],$$

where $m(x)$ is a codeword from the single-error-correcting BCH code of length 7, and $s(x)$ is a codeword from the double-error-correcting BCH code of length 7 (minimum distance 6), $i \in \{0, 1\}$ and $u(x) = (x^7 + 1)/(x + 1) = \sum_{j=0}^{6} x^j$.

Now let β be a primitive nth root of unity in GF(2^3). Thus β is a solution to $x^7 + 1 = 0$. The single-error-correcting BCH code has generator polynomial

$$g(x) = \text{LCM[minimum polynomials of } \beta, \beta^2].$$

The double-error-correcting BCH code has generator polynomial

$$g'(x) = \text{LCM[minimum polynomials of } 1, \beta, \beta^2, \beta^3, \beta^4].$$

Thus,

$$g(x) = \text{LCM}[x^3 + x + 1, x^3 + x + 1]$$
$$= x^3 + x + 1,$$

and,

$$g'(x) = \text{LCM}[x + 1, x^3 + x + 1, x^3 + x + 1, x^3 + x^2 + 1, x^3 + x + 1]$$
$$= x^7 + 1. \text{ (Thus, } s(x) = 0.)$$

By lemma 4.3 (a), $\psi(x) = x^7 + 1/g(x) = x^4 + x^2 + x + 1$. Also, since $\psi^2(x) = \psi(x)$, lemma 4.3 (a) reveals that $s = 0$. Then lemma 4.3 (b) gives

$$f(x) = x^n \psi(x) = x^4 + x^2 + x + 1, \text{ and}$$
$$q(x) \in \{0, 1, x, x^2, \ldots, x^6\}.$$

The vectors u of equation (4.2) are then formed by $[q(x), 0, q(x)f(x)]$ and there are eight such vectors seven of which are nonzero. The Nordstrom-Robinson code N'_{16} is formed by generating the code $\mathbf{K}_4$ which consists of all vectors of the form given by equation (4.3).

Observe that the Nordstrom-Robinson code can be interpreted as the set union of a linear code and of its seven cosets identified by the distinct u's for which $q(x) \neq 0$.

As a result of example 4.6 it can be seen that the Preparata codes are a generalization of the Nordstrom-Robinson codes. Preparata in [5] also shows that $\mathbf{K}_n$ has the largest number of codewords for its length and minimum distance, since it meets the Johnson bound [3] (see appendix A). In addition in [5], encoding and decoding methods are presented.

BIBLIOGRAPHY

[1] R.C. Bose and D.K. Ray-Chaudhuri, On a Class of Error Correcting Binary Group Codes, *Information and Control*, **3** (1960), 68-79.

[2] A. Hocquenghem, Codes correcteurs d'erreurs, *Chiffres* **2** (1959), 147-156.

[3] S.M. Johnson, A New Upper Bound for Error-Correcting Codes, *IEEE Transactions on Information Theory*, **8** (1962), 203-207.

[4] F.J. MacWilliams and N.J.A. Sloane, *The Theory of Error-Correcting Codes*, North-Holland Amsterdam, 1977.

[5] F.P. Preparata, A Class of Optimum Nonlinear Double-Error-Correcting Codes, *Information and Control*, **13** (1968), 378-400.

CHAPTER 5

The Vasil'yév Codes

INTRODUCTION

The Vasil'yév codes are a family of binary perfect (or *close packed*) single-error-correcting codes which contains both linear and nonlinear codes in its definition. The search for perfect codes was one of the earliest problems that arose in coding theory. In addition to being the best codes for their n and d, perfect codes are of interest to mathematicians, due mainly to their associated designs which we consider in Chapter 6.

PERFECT CODES

Recall that if we view a code of length n over a finite field F_q as a subset $\{x_1,x_2,\ldots,x_M\}$ of the vector space $V(F_q)$, the code is said to be perfect if for some integer t, the spheres of radius t around the M codewords completely fill $V(F_q)$ without overlap.

In the following theorem due to Hamming [1] we state the *sphere-packing* or *Hamming bound.*

Theorem 5.1

A q-ary (n,M,d) code, where $d = 2t + 1$ satisfies

$$M\left(\binom{n}{0} + \binom{n}{1}(q-1) + \cdots + \binom{n}{t}(q-1)^t\right) \leq q^n.$$

From this theorem it follows that a code which achieves the sphere-packing bound is perfect. Hence, for a perfect t-error-correcting code, the M spheres of radius t whose centers are the codewords fill the whole space $V(F_q)$ with no overlap.

For perfect single-error-correcting binary codes, theorem 5.1 gives

$$M\left(\binom{n}{0} + \binom{n}{1}\right) \leq 2^n. \tag{5.1}$$

The trivial perfect error-correcting codes are:

1) the binary repetition code

$$\begin{Bmatrix} 0\,0 \cdots 0 \\ 1\,1 \cdots 1 \end{Bmatrix},$$

2) a code with just one codeword, and

3) a code with all codewords in the space $(F_q)^n$ where F_q is an alphabet with q elements.

Two nontrivial perfect linear error-correcting codes are:

1) the $\left[n = \frac{q^r - 1}{q - 1}, n - r, 3\right]$ Hamming codes ($r \geq 2$), and

2) the binary [23,12,7] Golay code and the ternary [11,6,5] Golay code.

The Hamming codes are single-error-correcting, while the binary and ternary Golay codes are triple- and double-error-correcting, respectively.

It can be shown [5] that any code with the parameters of either of the two Golay codes must be equivalent to one of them. However, for single-error-correcting perfect codes a different situation exists. It was conjectured for some time that the Hamming codes and the binary and ternary Golay codes were the only nontrivial perfect codes. However, in 1962, J. L. Vasil'yév constructed a family of *nonlinear* perfect codes with the same parameters as the binary Hamming codes [6]. After that Schönheim [3] and Lindström [2] gave nonlinear codes with the same parameters as Hamming codes over GF(q) for any prime power q.

Thus the conjecture was weakened to: "any nontrivial perfect code has the same parame-

ters of a Hamming code or a Golay code." In 1973, A. Tietäväinen [4] provided a proof of this for q a prime power. We now proceed to Vasil'yév's constructive proof of the existence of a family of perfect single-error-correcting nonlinear codes.

CONSTRUCTION OF THE VASIL'YÉV CODES

We first state a few preliminary ideas. Let $n = 2^r - 1$, $r \geq 2$. Then by [1] the generator matrix of a perfect $[n,k,3]$ code has $k = n - r$ vectors of length n and the number of codewords is $2^k = 2^{n-r} = \frac{2^n}{2^r} = \frac{2^n}{n+1}$. Next let B_n be such a code containing the all zero vector. We write,

$$B_n = \{(\tau_1, \tau_2, \ldots, \tau_n)\},$$

where each $\tau_i \in \{0, 1\}$, $i = 1, 2, \ldots, n$. Let E^n be the set of all n-tuples over $\{0,1\}$. We write,

$$E^n = \{(\alpha_1, \ldots, \alpha_n)\},$$

where $\alpha_i \in \{0, 1\}$, $i = 1, 2, \ldots, n$. If $\tau \in B_n$ where $\tau = (\tau_1, \ldots, \tau_n)$, then let $\lambda(\tau)$ be an arbitrary function which equals either 0 or 1. That is, λ is any function which maps B_n to GF(2). Moreover, let $\lambda(0, \ldots, 0) = 0$ and let $|\alpha| = \alpha_1 + \cdots + \alpha_n$, where the + represents modulo 2 addition.

Now consider the code C formed by all $(2n + 1)$-component vectors of the form

$$c = (\alpha, \alpha + \tau, |\alpha| + \lambda(\tau)). \tag{5.2}$$

Theorem 5.2 [6]

The set of vectors C defined by equation (5.2) forms a perfect $[2^{r+1} - 1, 2n - r, 3]$ code containing the all zero codeword.

proof:

Clearly, C contains the all zero codeword. Also, the length of any codeword is $2n + 1 = 2(2^r - 1) + 1 = 2^{r+1} - 1$. Next, we verify that equation (5.1) is true.

The total number of possible vectors with $2n + 1$ components over GF(2) is 2^{2n+1}. This represents the right-hand side of equation (5.1). Next note that vectors having the form of equation (5.2) have $2n + 1$ components where there are $r + 1$ components that are dependent upon the others, that is, there are $r + 1$ redundant bits. Hence, there are $(2n + 1) - (r + 1) = 2n - r$ information bits. Consequently the left-hand side of equation (5.1) becomes

$$2^{2n-r}\left(1 + \binom{2n+1}{1}\right) = 2^{2n-r}(2n + 2)$$

$$= 2^{2n-r}\cdot 2(n + 1)$$

$$= 2^{2n-r}\cdot 2\cdot 2^{r}$$

$$= 2^{2n+1}.$$

In order to establish that the minimum distance of C is 3, let $\alpha, \beta \in E^n$ and $\tau, \nu \in B_n$ form the two codewords

$$c_1 = (\alpha, \alpha + \tau, |\alpha| + \lambda(\tau)),$$
$$c_2 = (\beta, \beta + \nu, |\beta| + \lambda(\nu)).$$

We will show that $d(c_1, c_2) \geq 3$, by considering two cases:

i) $\tau \neq \nu$,
ii) $\tau = \nu$.

In case i) with $\tau \neq \nu$, since $\tau, \nu \in B_n$, this implies that $d(\tau, \nu) \geq 3$. Then,

$$d(\alpha, \beta) = 0 \Rightarrow d(\alpha + \tau, \beta + \nu) \geq 3,$$
$$d(\alpha, \beta) = 1 \Rightarrow d(\alpha + \tau, \beta + \nu) \geq 2,$$
$$d(\alpha, \beta) = 2 \Rightarrow d(\alpha + \tau, \beta + \nu) \geq 1,$$
$$d(\alpha, \beta) = 3 \Rightarrow d(\alpha + \tau, \beta + \nu) \geq 0,$$

Hence, $d(c_1, c_2) \geq 3$.

In case ii) with $\tau = \nu$, α and β cannot be equal for otherwise $d(c_1, c_2) = 0$. Thus, with $\alpha \neq \beta$ consider two subcases $|\alpha| \neq |\beta|$ and $|\alpha| = |\beta|$. If $|\alpha| \neq |\beta|$, then

$$d(\alpha, \beta) \geq 1,$$
$$d(\alpha + \tau, \beta + \nu) \geq 1, \text{ and}$$
$$|\alpha| + \lambda(\tau) \neq |\beta| + \lambda(\nu).$$

This gives $d(c_1, c_2) \geq 3$. If $|\alpha| = |\beta|$, then

$$d(\alpha, \beta) \geq 2, \text{ and}$$
$$d(\alpha + \tau, \beta + \nu) \geq 2.$$

Thus, in this case $d(c_1, c_2) \geq 4$, which completes the proof.

Theorem 5.3 [6]

If $\lambda(\tau) = 0$, for all $\tau \in B_n$, then the Vasil'yév codes $C = \{(\alpha, \alpha + \tau, |\alpha| + \lambda(\tau)) : \alpha \in E^n, \tau \in B_n\}$ are perfect linear codes.

proof:

With λ identically equal to 0, codewords have the form $(\alpha, \alpha + \tau, |\alpha|)$ and clearly the sum of any two such codewords yields another codeword in C.

Theorem 5.4 [6]

If λ is a nonlinear function, then the Vasil'yév codes $C = \{(\alpha, \alpha + \tau, |\alpha| + \lambda(\tau)) : \alpha \in E^n, \tau \in B_n\}$ are perfect nonlinear codes.

proof:

This follows immediately by hypothesis, since there must exist $\tau_1, \tau_2 \in B_n$, such that $\lambda(\tau_1 + \tau_2) \neq \lambda(\tau_1) + \lambda(\tau_2)$.

Theorem 5.5

Any Vasil'yév code has the same parameters as some binary Hamming code.

proof:

Consider a Vasil'yév code V with length $2^{r+1} - 1$, $r \geq 2$. Note that this length always gives a value which is the length of a binary Hamming code H. The number of information bits in V is

$$\begin{aligned} 2n - r &= 2(2^r - 1) - r \\ &= 2^{r+1} - 2 - r \\ &= 2^{r+1} - 1 - (r + 1), \end{aligned}$$

which is equal to the number of information bits in H.

As a result of theorem 5.5, for a given n and d, the Vasil'yév codes are seen to be optimal as far as the number of codewords, M, is concerned.

In summary, it is seen that the Vasil'yév codes fill a gap that existed in the body of knowledge regarding perfect error-correcting codes. What remains open is:

1) the problem of finding all perfect codes that have the same parameters of the Hamming and Golay codes, and

2) the problem of finding all perfect codes over non-prime-power alphabets.

More about both of these problems will be mentioned in Chapter 8.

BIBLIOGRAPHY

[1] R.W. Hamming, Error Detecting and Error Correcting Codes, *Bell System Technical Journal* **29** (1950), 147-160.

[2] B. Lindström, On Group and Nongroup Perfect Codes in q Symbols, *Math. Scand.*, **25** (1969), 149-158.

[3] J. Schönheim, On Linear and Nonlinear Single-Error-Correcting q-nary Perfect Codes, *Information and Control,* **12** (1968), 23-26.

[4] A. Tietäväinen, On the Existence of Perfect Codes Over Finite Fields, *SIAM Journal of Applied Mathematics,* **24** (1973), 88-96.

[5] J. H. van Lint, Report of the Discrete Mathematics Group, Report 69-WSK-04 of the Technological University, Eindhoven, Netherlands (1969).

[6] J. L. Vasil'yév, On Nongroup Close-Packed Codes, *Problemy Kibernetiki,* **8** (1962), 337-339. English translation in *Problems in Cybernetics,* **8** (1965), 375-378.

CHAPTER 6

Designs and Nonlinear Codes

INTRODUCTION

Design theory has grown out of several areas in mathematics, in particular, statistics, graph theory, geometry, and coding theory. The emphasis here will be on the relationships between design theory and coding theory. It will be seen that it is always possible to obtain a t-design from a perfect code. Also known are sufficient conditions for the existence of t-designs associated with non-perfect codes.

DEFINITION OF t-DESIGN AND SOME SPECIAL CASES

Definition 6.1

A *t-design* or *tactical configuration* consists of a set X of v elements (called *points* or *varieties*) and a collection of distinct k-subsets of X (called *blocks*), with the property that any t-subset of X is contained in exactly λ blocks. The notation used to represent a t-design is t-(v,k,λ). Also, the number of blocks will be denoted by b.

The elements of the set X of a t-design are often called varieties because the original work with designs occurred in statistical experiments, especially in agriculture. For example, assume we have v varieties of fertilizer to be tested on b crops and that we are interested in the effects of pairs of fertilizers on the same crop. By using a t-design, each of the b crops can be tested with a block of k varieties of fertilizer, in such a way that each pair of varieties is tested together a constant number of λ times. Here then, $t = 2$ which gives a 2-(v,k,λ) design. 2-designs are often called *balanced incomplete block designs* (BIBD). If for a 2-design, $k = v$, then

it is termed a *complete design* or a *balanced block design* (BBD). If $v = b$ in a 2-design it is then a *symmetric design.*

Example 6.1

Suppose we have 7 fertilizers, that is, $X = \{1,2,\ldots,7\}$. Consider the following seven subsets of X:

$$\{1,2,4\},\ \{2,3,5\},\ \{3,4,6\},\ \{4,5,7\},\ \{5,6,1\},\ \{6,7,2\},\ \{7,1,3\}.$$

Here, $v = 7$, $b = 7$, $k = 3$, $\lambda = 1$. In order to see that $\lambda = 1$, take any pair of elements, and verify that one and only one of the seven subsets contains that pair. Hence, this 2-(7, 3, 1) design could be used to compare 7 fertilizers on 7 crops where each crop is given 3 fertilizers and any particular pair of fertilizers is applied to exactly one crop. Note in this example that the design is symmetric.

There is a simple geometrical representation of the design in example 6.1. The elements $1,\ldots,7$ can be represented as points and the blocks can be represented by lines (one being curved). The representation appears in figure 6.1

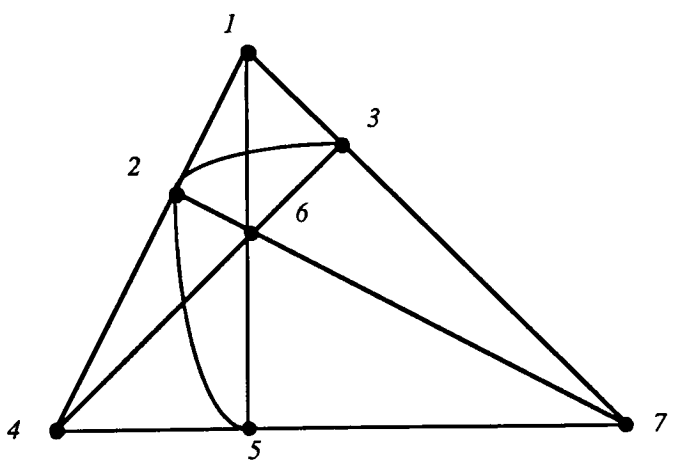


Figure 6.1

The 2-(7, 3, 1) design is also known as the *fano plane* or the *projective plane of order* 2. A projective plane is defined below.

Definition 6.2

A *projective plane* consists of points and lines satisfying:

a) every two points lie on exactly one line
b) every two lines intersect in exactly one point
c) every line contains at least three points
d) there are at least three points not on one line.

It can be shown that if some line in a projective plane has $n + 1$ points, then every line has $n + 1$ points and that there are a total of $n^2 + n + 1$ points. This number n is said to be the *order of the projective plane.* It is easy to see that a projective plane of order n is a 2-$(n^2+n+1, n+1, 1)$ design. The smallest projective plane which satisfies the requirements in definition 6.2 is that displayed in figure 6.1.

Another representation of the design in example 6.1 is obtained by indexing the rows and columns of a matrix with the blocks and points, respectively, of the design.

Definition 6.3

Given a t-(v, k, λ) design with v points $x_1, \ldots, x_v$ and b blocks $B_1, \ldots, B_b$, its $b \times v$ *incidence matrix* $A = (a_{ij})$ is defined by

$$a_{ij} = \begin{cases} 1 & \text{if } x_i \in B_j \\ 0 & \text{if } x_i \notin B_j. \end{cases}$$

Thus, the incidence matrix for the 2-(7,3,1) design in example 6.1 is

$$C = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{6.1}$$

A use of this incidence matrix is made in the next example where a code is constructed. Consideration of the technique will occur later in the chapter when *constant weight codes* are discussed.

Example 6.2

Consider C, the 7×7 incidence matrix from equation (6.1), along with the 7×7 matrix whose rows are the

complements of the rows of C. In addition, if we include the all zero vector 0000000 and the all one vector 1111111 we obtain the perfect single-error-correcting [7,4,3] Hamming code.

In example 6.1 there are $v = 7$ points, $b = 7$ blocks, $k = 3$ points per blocks, and $\lambda = 1$. If we let r = the number of blocks in which each element of X appears it is seen that $r = 3$. In general, r is called the *replication factor*. For any 2-(v,k,λ) design there are two relations that these five parameters satisfy.

Theorem 6.1 [4]

The parameters of a 2-(v,k,λ) design satisfy $bk = vr$ and $r(k-1) = \lambda(v-1)$.

proof:

The first equation can be thought of as counting the number of 1's in the incidence matrix in two different ways. There are b rows each with k 1's, and there are v columns each with r 1's. To prove the second equation, consider the b subsets of k elements each, and count the number of pairs containing a particular symbol θ. θ occurs in r sets and in each of these is paired with $k-1$ other symbols. However, θ must be paired with each of the $v-1$ exactly λ times.

It is worthwhile to note that the two conditions in theorem 6.1 are necessary but not sufficient for the existence of a BIBD.

Besides BIBD's which are t-designs with $t = 2$, another special t-design is now defined.

Definition 6.4

A *Steiner system* is a t-design with $\lambda = 1$. A t-$(v,k,1)$ design is often denoted by $S(t,k,v)$.

Clearly in example 6.1 is a Steiner system $S(2,3,7)$.

Example 6.3

a) Given the set of points $\{1,\ldots,9\}$ form the 3-sets below.

{1,2,3}	{1,4,7}	{1,5,9}	{1,6,8}
{4,5,6}	{2,5,8}	{2,6,7}	{2,4,9}
{7,8,9}	{3,6,9}	{3,4,8}	{3,5,7}

It can be seen that $v = 9$, $b = 12$, $k = 3$, $\lambda = 1$. This is the Steiner system $S(2,3,9)$. Also we can calculate that $r = \frac{bk}{v} = \frac{36}{9} = 4$, which of course, agrees with observation.

b) A partition of a set of ab elements into b sets of a elements each forms the Steiner system $S(1,a,ab)$.

In the next example a perfect code is seen to contain a Steiner system. Actually that is *always* the case which will be proven in theorem 6.2.

Example 6.4

The triple error-correcting [23,12,7] binary Golay code, G_{23}, has minimum distance 7 between any two codewords. Consider all the codewords of weight 7 in G_{23}. A counting argument shows that there are 253 such codewords. Form the 253×23 matrix of these codewords and observe that any set of four 1's will appear in one and only one row, for if the contrary were true for two rows r_1 and r_2, then the number of places in which r_1 and r_2 differ would be less than 7, an impossibility. Hence we have formed the incidence matrix of a t-design where $v = 23$, the number of blocks $b = 253$, the number of points in each block $k = 7$, such that any 4 points lie together in exactly one block ; that is, we have constructed the Steiner system $S(4,7,23)$.

MORE ON t-DESIGNS AND STEINER SYSTEMS

We now prove that is is always possible to obtain a Steiner system from a perfect code. This theorem is due to E.F. Assmus and H.F. Mattson.

Theorem 6.2 [1]

If there exists a perfect binary t-error-correcting code of length n, then there exists a Steiner system $S(t+1, 2t+1, n)$.

proof:

In the language of definition 6.1 we must show that there exists a set X of n points, and a collection of distinct $(2t+1)$-subsets (blocks) of X such that any $(t+1)$-subset of X lies in exactly one block. Consider the incidence matrix formed by all codewords of weight $2t+1$. Observe that any set of $t+1$ 1's appear in one and only one row, for if the contrary were true for any pair of rows then the number of places in which those two rows differ would be less that $2t+1$, which cannot be.

In theorem 6.2 there is no requirement of linearity in the perfect code. However, if linearity is imposed then the following stronger result can be proven.

Theorem 6.3 [1]

A linear code of length n, minimum distance $d = 2t+1$, and defined over GF(q) is perfect if and only if there exists a $(t+1)$-$\left(n, 2t+1, (q-1)^t\right)$ design.

Now we turn our attention to some properties of t-designs, in particular, various necessary conditions for t-designs to exist.

Theorem 6.4 [5]

Every t-(v,k,λ) design is also an i-(v,k,λ_i) design for $0 \leq i \leq t$, where

$$\lambda_i = \frac{\lambda \binom{v-i}{t-i}}{\binom{k-i}{t-i}}. \tag{6.2}$$

proof:

In definition 6.1 let the collection of blocks be denoted by $\boldsymbol{B}$ and let $\lambda(I) = |\{B \in \boldsymbol{B} : B \supseteq I\}|$. That is, $\lambda(I)$ is the number of blocks containing a given i-subset I of X. Now count in two ways the number of pairs (R,S) where R is a t-subset of X such that $I \subseteq R \subseteq S \in \boldsymbol{B}$. Given $I \subseteq X$, $|I| = i$, where $0 \leq i \leq t$, there are $\binom{v-i}{t-i}$ subsets R of X that contain I. Each is contained in λ blocks S. On the other hand, each block S of $\boldsymbol{B}$ that contains I must have $\binom{k-i}{t-i}$ t-subsets that contain I. Hence,

$$\lambda(I)\binom{k-i}{t-i} = \lambda\binom{v-i}{t-i}$$

Also, it follows that $\lambda(I)$ is independent of the i points originally chosen so that we may write $\lambda(I) = \lambda_i$, thus completing the proof.

We mention that the λ_i of equation (6.2) must, of course, be integers and that this is a necessary, but not sufficient condition for the existence of a t-design.

Corollary 6.5 [5]

The number of blocks in a t-design is

$$b = \lambda \frac{\binom{v}{t}}{\binom{k}{t}}.$$

proof:

The result follows from theorem 6.4 with $i = 0$ where $\lambda_0 = b$.

Theorem 6.6 [5]

In a t-(v,k,λ) design with b blocks and replication factor r

$$bk = vr.$$

proof:

Follows from corollary 6.5 and theorem 6.4 with $r = \lambda_1$.

It is possible to obtain new designs from existing designs. The next definition leads to a technique for doing so.

Definition 6.5

Let $P_1, \ldots, P_j$ be fixed points in a design. Consider the blocks containing $P_1, \ldots, P_j$, but not $P_{j+1}, \ldots, P_i$, for $0 \leq j \leq i \leq t$. The numbers of such blocks are called the *block intersection numbers*, and are denoted by λ_{ij}. If $j = 0$, we consider the blocks that *do not* contain $P_1, \ldots, P_i$ and if $j = i$ we consider the blocks which contain $P_1, \ldots, P_j$.

We list below, some properties of the block intersection numbers.

1) The λ_{ij} are well-defined for $0 \leq j \leq i \leq t$.

2) $\lambda_{00} = b$.

3) $\lambda_{ii} = \lambda_i$ for $i < t$ and $\lambda_{tt} = \lambda$.

4) The λ_{ij} satisfy a type of Pascal triangle property, namely,

$$\lambda_{ij} = \lambda_{i+1,j} + \lambda_{i+1,j+1}.$$

5) If the design is a Steiner system (i.e., $\lambda = 1$, then $\lambda_{tt} = \lambda_{t+1,t+1} = \cdots = \lambda_{kk} = 1$, and the λ_{ij} are defined for all $0 \leq j \leq i \leq k$.

Hence, for any t-(v,k,λ) design we can form the "Pascal triangle" of the associated block intersection numbers as displayed below in figure 6.2.

$i = 0$				λ_{00}				
$i = 1$			λ_{10}		λ_{11}			
$i = 2$		λ_{20}		λ_{21}		λ_{22}		
$i = 3$	λ_{30}		λ_{31}		λ_{32}		λ_{33}	
.	.		.		.		.	
.	.		.		.		.	
.	.		.		.		.	

Figure 6.2

Note that $\lambda_{00} = \lambda_0 = b$, $\lambda_{11} = \lambda_1$, $\lambda_{22} = \lambda_2, \cdots$ and that property (4) says that any λ_{ij} (e.g., λ_{21}) equals the sum of its immediate two immediate "neighbors" in the row below to the left and right (e.g., $\lambda_{21} = \lambda_{31} + \lambda_{32}$).

The calculation of the λ_{ij}'s proceeds by starting with $\lambda_{00} = b$. Next $\lambda_{11} = \lambda_1$ is calculated and then λ_{10}. The succeeding rows are handled in the same way by starting at the far right and working to the left element by element.

Example 6.5

Consider the 2-(7, 3, 1) design from example 6.1. From corollary 6.5,

$$\lambda_{00} = b = \frac{\lambda \begin{bmatrix} v \\ t \end{bmatrix}}{\begin{bmatrix} k \\ t \end{bmatrix}} = \frac{1 \begin{bmatrix} 7 \\ 2 \end{bmatrix}}{\begin{bmatrix} 3 \\ 2 \end{bmatrix}} = 7.$$

From theorem 6.4, $\lambda_{11} = \dfrac{1 \begin{bmatrix} 6 \\ 1 \end{bmatrix}}{\begin{bmatrix} 2 \\ 1 \end{bmatrix}} = 3$. By the Pascal triangle property, $\lambda_{10} = \lambda_{00} - \lambda_{11} = 4$. Similarly, $\lambda_{22} = 1$, $\lambda_{21} = 2$, and $\lambda_{20} = 2$. By property (5), $\lambda_{33} = \lambda_{22} = 1$, $\lambda_{32} = 0$, $\lambda_{31} = 2$, $\lambda_{30} = 0$. The result is

7

4 3

2 2 1

0 2 0 1

and any entry is a λ_{ij} which can thus be interpreted accordingly. For example, $\lambda_{21} = 2$. Observe λ_{21} is the number of blocks in the design that contain P_1, but do not contain P_2. Choosing the assignment $P_1 = 2$, $P_2 = 3$, $P_3 = 5$, and considering figure 6.1, it is seen that there are two such blocks, namely 124 and 672.

Given a t-(v, k, λ) design with block intersection numbers λ_{ij}, we can derive other designs from it. Suppose we let $\mathbf{B}$ be the set of all b blocks of a t-design. Now delete one of the v points, say P_1, from the design. Consider the blocks that remain - there are two sets:

1) a set $\mathbf{B}_1$ which contains the λ_{10} blocks from the original v points that did not contain P_1 to begin with, and

2) a set $\mathbf{B}_2$ which contains the λ_{11} blocks that included, before deletion, the point P_1.

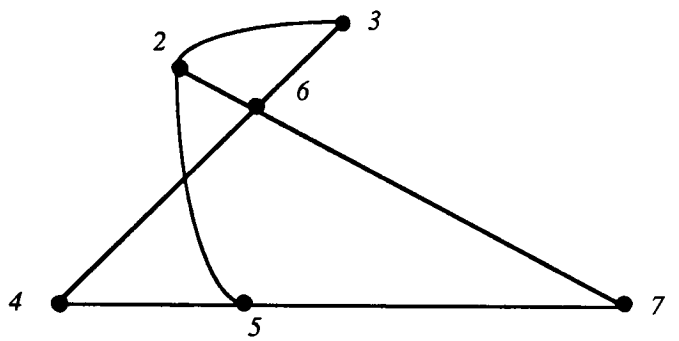
Theorem 6.7 [5]

a) The blocks $\mathbf{B}_1$ form a $(t-1)$-$(v-1,k,\lambda_{t,t-1})$ design with block intersection numbers $\lambda_{ij} = \lambda_{i+1,j}$.

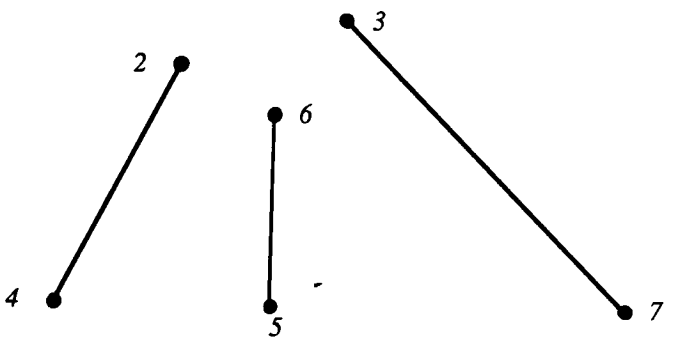
b) The blocks $\mathbf{B}_2$ form a $(t-1)$-$(v-1,k,\lambda)$ design with block intersection numbers $\lambda_{ij} = \lambda_{i+1,j+1}$).

Example 6.6

Delete 1 from figure 6.1 of the 2-(7, 3, 1) design. We display the blocks that remain.



Shown above is the set $\mathbf{B}_1$, the set of $\lambda_{10} = 4$ blocks that do not contain 1. Since 1 has been deleted there are now $v - 1 = 6$ points. Also since any one point belongs to 3 blocks we see $b - r = 4$ blocks remaining. In general, from the Pascal triangle property, there will be $\lambda_{00} - \lambda_{11} = \lambda_{10}$ blocks that remain. These blocks are still 3-subsets and we note that each $t - 1 =$ 1-subset is contained in exactly $\lambda_{21} = 2$ blocks. Thus, we have a 1-(6, 3, 2) design. Now consider the blocks $\mathbf{B}_2$ displayed below.



These are the $\lambda_{11} = 3$ blocks with $v - 1 = 6$ points, with a block size of $k - 1 = 2$ and any $t - 1 =$ 1-subset is contained in exactly $\lambda = 1$ blocks. Consequently, we have a 1–(6, 2, 1) design.

An obvious corollary follows as a consequence of part (b) of theorem 6.7

Corollary 6.8

If a Steiner system $S(t,k,v)$ exists, then it follows that $S(t-1,k-1,v-1)$ is also a Steiner system.

DESIGNS FROM CODES

Clearly, design construction is not easy. However, it has been proven that a 2-design exists if v is sufficiently large for fixed k and λ [12,13,14].

It is also possible to obtain 2-designs and 3-designs from Hadamard matrices.

Theorem 6.9 [3]

If a Hadamard matrix of order $n > 4$ exists then there exists a symmetric 2-$(n-1, \frac{1}{2}n-1, \frac{1}{4}n-1)$ design called a *Hadamard 2-design*, and conversely.

We partially illustrate the method of proof with an example.

Example 6.7

Consider the Hadamard matrix H_8 displayed below.

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix}$$

Deleting the first row and first column and then replacing -1 by 0 throughout gives the remaining 7×7 matrix below which is seen to be the incidence matrix of a symmetric 2-(7, 3, 1) design.

$$\begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}$$

Theorem 6.10 [3]

If a Hadamard matrix of order $n > 4$ exists then there exists a 3-$(n, \frac{1}{2}n, \frac{1}{4}n-1)$ design called a *Hadamard 3-design.*

Again we present the method of construction with an example.

Example 6.8

Using H_8 from example 6.7, any row except the first has $\frac{1}{2}n$ components $+1$ and $\frac{1}{2}n$ components -1. Taking the columns with $+1$ as points (-1 will work as well) and the rows as blocks, then rows 2-8 of H_8 give the subsets

$$\{1,3,5,7\},\ \{1,2,5,6\},\ (1,4,5,8\},\ \{1,2,3,4\},\ \{1,3,6,8\},\ \{1,2,7,8\},\ \{1,4,6,7\}$$

which form a 3-$(8,4,1)$ design.

Another way to derive designs from codes is through the use of *uniformly packed codes* which we define next. Let E^n stand for the set of all n-tuples over some alphabet.

Definition 6.6

For an (n,M,d) code C, consider the set $Y \subset E^n$, such that for any $y \in Y$ and $c \in C$ the distance $d(y,c) \geq e$. Let $r(y)$ denote the number of vectors c in C such that $d(y,c)$ equals e or $e+1$. A code C where $r(y) = r$ = constant for all $y \in Y$ is called a *uniformly packed code of parameter r.*

Examples of uniformly packed codes include certain Hadamard codes, Reed-Muller codes, and the Preparata codes. The significance of definition 6.6 is that Semokov, Zinov'ev and Zaitsev in [11] show that for a uniformly packed code with the zero vector, the set of codewords of any one weight *always* form a t-design where the parameters are completely defined by the code parameters.

In 1969 Assmus and Mattson gave an important sufficient condition on a code, which is not necessarily perfect, for the existence of associated t-designs. The details of this can be found in [9, chapter 6] or in Assmus and Mattson [2]. Many new 5-designs have been obtained in this fashion. It was a long-standing conjecture that t-designs with $t \geq 6$ did not exist. However the discovery of the 6-$(33,8,36)$ design was first announced in 1984 by Magliveras and Leavitt [8, pp.337-352]. Shortly thereafter, Kramer, Leavitt, and Magliveras discovered another 6-design, a 6-$(20,9,112)$ [7]. The third 6-design a 6-$(14,7,4)$ was discovered by Kreher and Radziszowski [6] and could be considered the most interesting so far since it is the smallest simple 6-design that can exist.

NONLINEAR CODES FROM DESIGNS

Before leaving the topic of designs and codes we briefly consider *constant weight codes*, that is, codes where each codeword has the same weight. In order to obtain such a code, suppose we have a Steiner system $S(t,k,v)$. The rows of the incidence matrix form an (n,M,d) code C, where clearly $n = v$, $M = b = \dfrac{\binom{v}{t}}{\binom{k}{t}}$. What is the minimum distance for codewords in C? Observe that because we have a Steiner system (i.e., $\lambda = 1$), codewords of C have no more than $t - 1$ points (1's) in common. Thus with every codeword having constant weight k, the number of coordinates that are different $\geq 2(k - t + 1)$. Thus $d \geq 2(k - t + 1)$.

Constant weight codes, formed as such are, in general, nonlinear and have been studied by N.V. Semakov and V.A. Zinov'ev [10]. In chapter 7 we shall return to the topic of constant weight codes.

BIBLIOGRAPHY

[1] E.F. Assmus Jr. and H.F. Mattson Jr., On Tactical Configurations and Error-Correcting Codes, *Journal of Combinatorial Theory*, **2** (1967) 243-257.

[2] E.F. Assmus Jr. and H.F. Mattson Jr., Coding and Combinatorics, *SIAM Review*, **16** (1974) 349-388.

[3] V.N. Bhat and S.S. Shrikhande, Non-isomorphic Solution of Some Balanced Incomplete Block Designs, *Journal of Combinatorial Theory*, **9A** (1970), 174-191.

[4] R.C. Bose and K.R. Nair, Partially Balanced Incomplete Block Designs, *Sankhyā*, **4** (1939) 337-372.

[5] D.R. Hughes, On t-designs and Groups, *American Journal of Mathematics*, **87** (1965) 761-778.

[6] D.L. Kreher and S.P. Radziszowski, The Existence of Simple 6-(14, 7, 4) Designs, *Journal of Combinatorial Theory)*, **A** (submitted March 1986).

[7] E.S Kramer, D.W. Leavitt and S.S. Magliveras, Construction Procedures for t-designs and the Existence of New Simple 6-designs, *Annals of Discrete Mathematics*, **26** (1985) 247-274.

[8] S.S. Magliveras and D.W. Leavitt, Simple 6–(33, 8, 36) Designs from $PGL_2(32)$, *Computational Group Theory, Proceedings of the London Mathematical Society Symposium on Computational Group Theory*, Academic Press, 1984.

[9] F.J. MacWilliams and N.J.A. Sloane, *The Theory of Error-Correcting Codes*, North-Holland Amsterdam, 1977.

[10] N.V. Semakov and V.A. Zinov'ev, Balanced Codes and Tactical Configurations, *Problems of Information Transmission*, **5** (1969) 22-28.

[11] N.V. Semakov, V.A. Zinov'ev, and G.V. Zaitsev, Uniformly Packed Codes, *Problems of Information Transmission*, **7** (1971), 30-39.

[12] R.M. Wilson, An Existence Theory for Pairwise Balanced Designs, *Journal of Combinatorial Theory,* **13A** (1972) 220-273.

[13] R.M. Wilson, The Necessary Conditions for t-designs are Sufficient for Something, *Utilitas Math.,* **4** (1973) 207-215.

[14] R.M. Wilson, An Existence Theory for Pairwise Balanced Designs: III - Proof of the Existence Conjectures, *Journal of Combinatorial Theory,* **18A** (1975) 71-79.

CHAPTER 7

A Method of Finding Codes Via Computer Search

INTRODUCTION

As mentioned at the outset of this thesis, one advantage of a nonlinear code over a linear code is that the former often have a higher information rate. In general a "good" (n,M,d) code has small n (for fast transmission of messages), large M (to enable transmission of a wide variety of messages), and large d (to correct many errors). However, these are conflicting aims and thus in practice the problem may be reduced to finding the largest code (i.e., the largest M) for a given length n and minimum distance d. Which n and d might be chosen is the subject of a paper by E.R. Berlekamp [1] where the relationship of cost of encoders and decoders is compared to the important system parameters, speed and delay.

In any case let us suppose that n and d have been chosen and we wish to find the largest M via an exhaustive computer search. For example, consider a binary code with $n = 10$ and $d = 3$. It is known that $72 \leq A(10,3) \leq 79$ [2]. An attempt to tighten this bound by an exhaustive search for a (10,73,3) binary code would involve a calculation of d for each pair of vectors in each of the $\binom{2^{10}}{73}$ subsets of the 2^{10} binary 10-tuples. Clearly, this approach is not feasible. In the next section an alternative idea is explored.

A COMPUTER SEARCH FOR NEW CODES

Suppose we consider the collection of all binary n-tuples. Clearly, this is a set of 2^n vectors and we might envision a $2^n \times 2^n$ matrix whose rows and columns are indexed by these vectors v_i, $i = 1, \ldots, n$, where the entry in the ith row and jth column is the Hamming distance $d(v_i, v_j)$. If we could efficiently store and search such a matrix for a subset of size M vectors with minimum distance d we would have a straight-forward way of obtaining an (n,M,d) code with the capability of correcting $\left\lfloor \frac{d-1}{2} \right\rfloor$ errors.

The difficulty that arises is, when n is as small as ten or more, the matrix of Hamming distances has 10^6 or more entries, and thus both storage and searching become a problem. In this section we show a method for "compressing" the information in the $2^n \times 2^n$ matrix of Hamming distances into a smaller matrix. In doing so, it will be seen that we will lose some generality and detail, but we will gain by having constructed a significantly smaller matrix that needs to be stored and searched. The eventual result as we shall see is that it should be possible to discover new nonlinear error-correcting codes, some of which are cyclic.

First we cite a definition which will lead to the method just mentioned.

Definition 7.1

Let G be any permutation group on the set $\{1, \ldots, n\}$. For any $i \in \{1, \ldots, n\}$ let $i^G = \{i^g : g \in G\}$. The set i^G is called the *orbit of i under G*.

In words, the orbit of i under G is a set containing all the elements in $\{1, \ldots, n\}$ to which i maps. A simple example follows.

Example 7.1

Consider the set $G = \{(1), (132)(465)(78), (132)(465), (123)(456), (123)(456)(78), (78)\}$ which is a permutation group on $\{1, \ldots, 8\}$. Then,

$1^G = \{1,2,3\}$ $\quad$ $5^G = \{5,4,6\}$

$2^G = \{2,1,3\}$ $\quad$ $6^G = \{6,5,4\}$

$3^G = \{3,2,1\}$ $\quad$ $7^G = \{7,8\}$

$4^G = \{4,6,5\}$ $\quad$ $8^G = \{8,7\}$.

We next extend the idea of the orbit of a single member of a set $\{1, \ldots, n\}$ with the following definition.

Definition 7.2

Let G be any permutation group on the set $X = \{1, \ldots, n\}$. For any $S \subseteq X$ and $g \in G$ let $S^g = \{s^g : s \in S\}$. The set S^g is called the *orbit of the set S under g* .

We note that as elements of G act on the power set of the set $X = \{1, \ldots, n\}$ or on the collection of t-subsets of X, the effect is to produce a collection of orbits of sets. Shortly, we will see that these orbits of sets can be thought of as *orbits of codewords* .

Example 7.2

Let $X = \{1, 2, 3, 4\}$ and consider all bijections which map X onto itself. We have

$f_1 = (1)$	$f_7 = (12)$	$f_{13} = (132)$	$f_{19} = (1432)$
$f_2 = (34)$	$f_8 = (12)(34)$	$f_{14} = (1342)$	$f_{20} = (142)$
$f_3 = (23)$	$f_9 = (123)$	$f_{15} = (13)$	$f_{21} = (143)$
$f_4 = (234)$	$f_{10} = (1234)$	$f_{16} = (134)$	$f_{22} = (14)$
$f_5 = (243)$	$f_{11} = (1243)$	$f_{17} = (13)(24)$	$f_{23} = (1423)$
$f_6 = (24)$	$f_{12} = (124)$	$f_{18} = (1324)$	$f_{24} = (14)(23)$

Now let $G = \{f_1, \ldots, f_{24}\}$ and let a binary operation be function composition o. Then, as in example 3.1, $[G, o]$ is a group of permutations. It is easy to see that $1^G = 2^G = 3^G = 4^G = \{1, 2, 3, 4\}$. Now use definition 7.2 to determine the orbit of $\{1, 2\} \subseteq X$ under $f_3 = (23)$. Obviously, $\{1, 2\}^{f_3} = \{1, 3\}$. If we produce all the orbits of the set $\{1, 2\}$ under $g \in G$ we obtain:

$\{1,2\}^{f_1} = \{1,2\}$	$\{1,2\}^{f_7} = \{2,1\}$	$\{1,2\}^{f_{13}} = \{3,1\}$	$\{1,2\}^{f_{19}} = \{4,1\}$
$\{1,2\}^{f_2} = \{1,2\}$	$\{1,2\}^{f_8} = \{2,1\}$	$\{1,2\}^{f_{14}} = \{3,1\}$	$\{1,2\}^{f_{20}} = \{4,1\}$
$\{1,2\}^{f_3} = \{1,3\}$	$\{1,2\}^{f_9} = \{2,3\}$	$\{1,2\}^{f_{15}} = \{3,2\}$	$\{1,2\}^{f_{21}} = \{4,2\}$
$\{1,2\}^{f_4} = \{1,3\}$	$\{1,2\}^{f_{10}} = \{2,3\}$	$\{1,2\}^{f_{16}} = \{3,2\}$	$\{1,2\}^{f_{22}} = \{4,2\}$
$\{1,2\}^{f_5} = \{1,4\}$	$\{1,2\}^{f_{11}} = \{2,4\}$	$\{1,2\}^{f_{17}} = \{3,4\}$	$\{1,2\}^{f_{23}} = \{4,3\}$
$\{1,2\}^{f_6} = \{1,4\}$	$\{1,2\}^{f_{12}} = \{2,4\}$	$\{1,2\}^{f_{18}} = \{3,4\}$	$\{1,2\}^{f_{24}} = \{4,3\}$

However, let us agree to list only those orbits of a set which are cyclic shifts of its members in $X = \{1, \ldots, n\}$. Thus, the orbits of $\{1, 2\}$ are $\{1, 2\}$, $\{2, 3\}$, $\{3, 4\}$, and $\{4, 1\}$. Similarly, the set $\{1, 3\}$ in $\{1, 2, 3, 4\}$ has as orbits the sets $\{1, 3\}$ and $\{2, 4\}$.

A complete list of all orbits of the power set of $X = \{1, 2, 3, 4\}$ follows.

$\binom{4}{1}$ orbits with 1 element	$\binom{4}{2}$ orbits with 2 elements		$\binom{4}{3}$ orbits with 3 elements
1	12	13	123
2	23	24	234
3	34		134
4	14		124
Orbit of {1}	Orbit of {1,2}	Orbit of {1,3}	Orbit of {1,2,3}

We also could have listed the empty set as its own orbit and X as its own orbit.

Now let us relate the ideas in example 7.2 to the construction of codes. Notice from example 7.2 that any member of the power set of $\{1, 2, 3, 4\}$ can be considered to be a binary 4-tuple where the inclusion of a decimal digit is equivalent to the corresponding bit in that position being a 1 (numbering, say, from the left and starting at 1). Hence, if we rewrite the above list of orbits we obtain the following list of orbits of 1-, 2-, and 3-tuples.

$\binom{4}{1}$ 4-tuples with 1 element	$\binom{4}{2}$ 4-tuples with 2 elements		$\binom{4}{3}$ 4-tuples with 3 elements
1000	1100	1010	1110
0100	0110	0101	0111
0010	0011		1011
0001	1001		1101
Orbit of 1000 Denoted by Δ_1	Orbit of 1100 Denoted by Δ_2	Orbit of 1010 Denoted by Δ_3	Orbit of 1110 Denoted by Δ_4

Also we define $0000 = \Delta_0$ and $1111 = \Delta_5$. Now form a matrix indexed by the Δ_i, $i = 0,1,\ldots,5$, where the entry in the ith row and jth column is denoted by $d(\Delta_i,\Delta_j)$ and is determined from

$$d(\Delta_i,\Delta_j) = \min \{d(u,v) : u \in \Delta_i, v \in \Delta_j \text{ and if } i = j, \text{ then } u \neq v\}. \tag{7.1}$$

	Δ_0	Δ_1	Δ_2	Δ_3	Δ_4	Δ_5
Δ_0	∞	1	2	2	3	4
Δ_1	1	2	1	1	2	3
Δ_2	2	1	1	2	1	2
Δ_3	2	1	2	4	1	2
Δ_4	3	2	1	1	2	1
Δ_5	4	3	2	2	1	∞

We formalize this idea with the following definition.

Definition 7.3

The matrix $D = \left(d(\Delta_i,\Delta_j)\right)$ is called the *reduced matrix of Hamming distances.*

The example which follows now uses the reduced matrix of Hamming distances to construct codes.

Example 7.3

Two simple examples of single-error-correcting codes are:

a) the code formed by the union of Δ_0 with Δ_5 is a $(4, 2, 4)$ code, and

b) the code formed by Δ_3 is a $(4, 2, 4)$ code.

It is easily seen that the code in (a) is linear and the code in (b) is nonlinear and cyclic.

The two previous examples suggest that what we have been calling orbits of n-tuples may also be called *orbits of codewords* and can be used to construct error-correcting codes. Moreover, in order to determine the minimum distance between codewords we can effect a savings of space by indexing the matrix of Hamming distances with the orbits of codewords rather than the individual codewords themselves.

When determining the minimum Hamming distance between two orbits we have used equation (7.1) which requires checking all possible pairs of vectors in the two orbits in question. The next two theorems reveal more about this notion.

Theorem 7.1

If G is a group of permutations on $\{1, \ldots, n\}$ and $g \in G$, then $d(u,v) = d(u^g, v^g)$ where $u \in \Delta_i$ and $v \in \Delta_j$ for all i, j.

proof:

Applying the same permutation to both u and v leaves the same coordinate positions of u paired up with the same coordinate positions of v and thus $d(u,v) = d(u^g, v^g)$.

Theorem 7.2

If G is a group of permutations on $\{1, \ldots, n\}$ and $g \in G$, then $\min \{d(u_0, v)\} = \min \{d(u_0, v^g)\}$ where $u_0 \in \Delta_i$ and fixed, and $v \in \Delta_j$ for all i, j.

proof:

With u_0 fixed, the value of $\min \{d(u_0, v)\}$ is found when $v \in \Delta_j$ is a vector with a maximum number of coordinate positions having 1's in common with the corresponding coordinate positions of u_0. However, since v^g is just another vector in Δ_j, $\min \{d(u_0, v)\} = \min \{d(u_0, v^g)\}$.

Theorems 7.1 and 7.2 imply then,

$$d(\Delta_i, \Delta_j) = \min \{d(u, v)\} = \min \{d(u_0, v)\},$$

where $u \in \Delta_i$, $v \in \Delta_j$, and $u_0 \in \Delta_i$ (and fixed).

Example 7.4

Let us construct a constant weight 4 binary code of length 7. We begin by listing the $\binom{7}{4} = 35$ different 7-tuples. Also, for brevity we use decimal digits to indicate which bits are 1's.

1234	1235	1236	1245	1357
2345	2346	2347	2356	1246
3456	3457	1345	3467	2357
4567	1456	2456	1457	1346
1567	2567	3567	1256	2457
1267	1367	1467	2367	1356
1237	1247	1257	1347	2467
Orbits of 1111000 Δ_0	Orbits of 1110100 Δ_1	Orbits of 1110010 Δ_2	Orbits of 1101100 Δ_3	Orbits of 1010101 Δ_4

The matrix indexed by Δ_i, $i = 0, \ldots, 4$ is

	Δ_0	Δ_1	Δ_2	Δ_3	Δ_4
Δ_0	2	2	2	2	2
Δ_1	2	4	2	2	2
Δ_2	2	2	4	2	2
Δ_3	2	2	2	2	2
Δ_4	2	2	2	2	2

Hence, we obtain two binary single-error-correcting constant weight 4 codes, one given by Δ_1, and the other given by Δ_2. Each is a (7, 7, 4) cyclic code.

In order to formalize the ideas herein presented so that computer searching might be useful we now assume that for a given n and d we have constructed a reduced matrix of Hamming distances as given by definition 7.3.

Definition 7.4

The matrix $M = (m_{ij})$, where

$$m_{ij} = \begin{cases} 1, & d(\Delta_i, \Delta_j) \geq d \\ 0, & \text{otherwise} \end{cases}$$

is called the *incidence Hamming distance matrix* .

With this definition the problem of finding a collection of codewords with length n and minimum distance d then becomes equivalent to finding a square submatrix in the incidence Hamming distance matrix M all of whose entries are 1's. For example, suppose that the matrix M with rows and columns indexed by some $\Delta_0, \ldots, \Delta_4$ is

$$M = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Then, in this case there *is* a square submatrix of all 1's that is indexed by Δ_1 and Δ_3. This implies that the code formed by the union of Δ_1 with Δ_3 is a code capable of correcting $\left\lfloor \frac{d-1}{2} \right\rfloor$ errors.

We note that in order to find a submatrix of all 1's requires that we find a clique in the square submatrix of M whose diagonal elements are 1's and hence, in general, this problem is NP-complete. Thus a backtracking algorithm would be appropriate for a computer search in this situation.

BIBLIOGRAPHY

[1] E.R. Berlekamp, The Technology of Error-Correcting Codes, *Proceedings of the IEEE*, **68** (1980), 546-593.

[2] N.J.A. Sloane, Recent Bounds for Codes, *Contemporary Mathematics* **9** (1982), 153-185.

CHAPTER 8

Concluding Remarks

INTRODUCTION

We have seen that the study of error-correcting codes which began with Richard Hamming in the late 1940's is a highly mathematical area of computer science. Nonlinear codes, the topic of this thesis, have been constructed by utilizing matrix theory (Hadamard codes), group theory (Nordstrom-Robinson codes), Galois fields (Preparata codes), and juxtaposition of codes (Vasil'yév codes). Moreover, some suggestions were made regarding the discovery of new codes via a computer search.

COMMENTS REGARDING CODESIZE

As mentioned at the outset, one advantage of a nonlinear code over a linear code is that the former often have a higher information rate. This, of course, happens because it is often possible to construct a nonlinear code with more codewords than the best linear code with the same n and d. Thus we have seen as a recurring notion, the problem of finding codes which are optimal as far as M is concerned.

For any q-ary (n,M,d) code, if given the values of n, q, and d, the sphere-packing bound provides an upper bound on $A(n,d)$. In the case of binary codes, the sphere-packing bound is reasonably good for cases where $n \geq 2d + 1$. However, it does not work as well when $n < 2d$. Fortunately, in this case there is a tighter bound, namely the Plotkin bound. And, as observed, in chapter 2 there are codes that meet the Plotkin bound for at least some values of n and d.

The study of the values of $A(n,d)$ is considered by many to be the central problem of coding theory, however, $A(n,d)$ is rarely known exactly unless n and d are relatively small, or $n < 2d$. Typically, researchers have sought to find upper and lower bounds for $A(n,d)$.

If n and d are relatively small, lower bounds may be obtained by exhibiting a set of M vectors with the given n and d, thus implying $A(n,d) \geq M$. In order to obtain an upper bound the usual approach is to rely on linear programming techniques whereby a linear function is maximized (or minimized) subject to a set of constraints. We will not pursue that topic any further here, but rather the reader is referred to [4, chapter 17].

Nonlinear codes have, in the last ten years, experienced a renewed popularity. As a result of this, the well-known table of best known codes in MacWilliams and Sloane [4, p.674] has many places in it where updating can occur. Appendix A contains the necessary changes in table I which is a table of the best known binary codes with $n \leq 24$ and $d \leq 10$ that was provided directly to this author by N.J.A. Sloane [5]. This table along with many details of the associated research will appear in the soon to be published book [2]. In that work there will also appear many references to updated tables for the best known constant weight codes and, in addition, this author has learned that T. Verhoeff at the University of Technology in Eindhoven, Netherlands is presently preparing an updated list of the best known binary linear codes [5].

SHORTCOMINGS

In this thesis the matters of encoding and decoding have not been addressed. Here we have taken the stance that a "good" code is one which, for a given n and d, has maximum M.

Another shortcoming herein is that no new codes were found, although through discussions with Professor Donald Kreher, it was possible to formulate the basis by which a computer search for new codes could be made (see chapter 7).

No consideration of coding complexity is made in this thesis. However, a general reference is [3].

FUTURE WORK

Any of the shortcomings mentioned above could be used as a springboard for future work in the area of nonlinear codes. Probably the most exciting area would be to use the discussion of new codes in chapter 7 as a starting point for a thesis at this level.

Another possible direction could be to prove or disprove the existence of the Hadamard matrix H_{428}. The existence of H_{428} would be demonstrated if it were possible to find a Williamson matrix of order 107.

In chapter 5 two open questions were mentioned:

1) the problem of finding all perfect codes that have the same parameters of the Hamming and Golay codes, and

2) the problem of finding all perfect codes over non-prime-power alphabets.

Either of these is also a candidate for future work. As far as question 2 is concerned, it has been conjectured by Best [1] that there are no nontrivial perfect codes over non-prime-power alphabets, but of course the question is still open.

BIBLIOGRAPHY

[1] M.R. Best, Perfect Codes Hardly Exist, *IEEE Transactions on Information Theory*, **29** (1983), 349-351.

[2] J.H. Conway and N.J.A. Sloane, *Sphere Packings, Lattices and Groups*, Springer-Verlag, New York, 1987.

[3] G. Longo, ed., *Coding and Complexity*, Springer-Verlag, New York, 1975.

[4] F.J. MacWilliams and N.J.A. Sloane, *The Theory of Error-Correcting Codes*, North-Holland Amsterdam, 1977.

[5] N.J.A. Sloane, Private Communication, August 13, 1987.

APPENDIX A

A Table of Values of $A(n,d)$

n	$d=4$	$d=6$	$d=8$	$d=10$
6	4	2	1	1
7	8	2	1	1
8	16	2	2	1
9	20	4	2	1
10	40	6	2	2
11	72-79	12	2	2
12	144-158	24	4	2
13	256	32	4	2
14	512	64	8	2
15	1024	128	16	4
16	2048	256	32	4
17	2720-3276	256-340	36-37	6
18	5120-6552	512-680	64-74	10
19	10240-13104	1024-1288	128-144	20
20	20480-26208	2048-2372	256-279	40
21	36864-43690	2560-4096	512	40-48
22	73728-87380	4096-6942	1024	48-88
23	147456-173784	8192-13774	2048	64-150
24	294912-344636	16384-24106	4096	128-280

This table gives values of $A(n,d)$, the largest number, M, of codewords in an (n,M,d) code. By theorem 1.3 it is sufficient to show just the even values of d (or just the odd values of d). Where one number is listed the value is known exactly, otherwise the values are upper and lower bounds for $A(n,d)$.

BIBLIOGRAPHY

The bibliography which follows contains all references in alphabetical order from chapters 1-8 . A number in square brackets following the reference indicates the chapter in which that reference appears.

E.F. Assmus Jr. and H.F. Mattson Jr., On Tactical Configurations and Error-Correcting Codes, *Journal of Combinatorial Theory,* **2** (1967) 243-257. [6]

E.F. Assmus Jr. and H.F. Mattson Jr., Coding and Combinatorics, *SIAM Review,* **16** (1974) 349-388. [6]

E.R. Berlekamp, The Technology of Error-Correcting Codes, *Proceedings of the IEEE,* **68** (1980), 546-593. [7]

M.R. Best, Perfect Codes Hardly Exist, *IEEE Transactions on Information Theory,* **29** (1983), 349-351. [8]

V.N. Bhat and S.S. Shrikhande, Non-isomorphic Solution of Some Balanced Incomplete Block Designs, *Journal of Combinatorial Theory,* **9A** (1970), 174-191. [6]

R.C. Bose and K.R. Nair, Partially Balanced Incomplete Block Designs, *Sankhyā*, **4** (1939) 337-372. [6]

R.C. Bose and D.K. Ray-Chaudhuri, On a Class of Error Correcting Binary Group Codes, *Information and Control,* **3** (1960), 68-79. [4]

R.C. Bose and S.S. Shrikhande, A Note on a Result in the Theory of Code Construction, *Information and Control,* **2** (1959), 183-194. [2]

J.H. Conway and N.J.A. Sloane, *Sphere Packings, Lattices and Groups,* Springer-Verlag, New York, 1987. [8]

A.W. Geramita and J. Seberry, Orthogonal Designs, *Lecture Notes in Pure and Applied Mathematics,* N 45 [2]

J.M. Goethals, On the Golay Perfect Binary Code, *Journal of Combinatorial Theory,* **11** (1971), 178-186. [3]

R.W. Hamming, Error Detecting and Error Correcting Codes, *Bell System Technical Journal,* **29** (1950), 147-160. [1,5]

A. Hocquenghem, Codes correcteurs d'erreurs, *Chiffres* **2** (1959), 147-156. [4]

D.R. Hughes, On t-designs and Groups, *American Journal of Mathematics,* **87** (1965) 761-778. [6]

S.M. Johnson, A New Upper Bound for Error-Correcting Codes, *IEEE Transactions on Information Theory,* **8** (1962), 203-207. [3,4]

E.S Kramer, D.W. Leavitt and S.S. Magliveras, Construction Procedures for t-designs and the Existence of New Simple 6-designs, *Annals of Discrete Mathematics,* **26** (1985) 247-274. [6]

D.L. Kreher and S.P. Radziszowski, The Existence of Simple 6-(14, 7, 4) Designs, *Journal of Combinatorial Theory),* **A** (submitted March 1986). [6]

V.I. Levenshtein, The Application of Hadamard Matrices to a Problem in Coding, *Problemy Kibernetiki,* **5** (1961), 123-136. English translation in *Problems in Cybernetics,* **5** (1964), 166-184. [2]

B. Lindström, On Group and Nongroup Perfect Codes in q Symbols, *Math. Scand.,* **25** (1969), 149-158. [5]

G. Longo, ed., *Coding and Complexity,* Springer-Verlag, New York, 1975. [8]

F.J. MacWilliams and N.J.A. Sloane, *The Theory of Error-Correcting Codes,* North-Holland Amsterdam, 1977. [1,2,3,4,6,8]

S.S. Magliveras and D.W. Leavitt, Simple 6–(33, 8, 36) Designs from $PGL_2(32)$, *Computational Group Theory, Proceedings of the London Mathematical Society Symposium on Computational Group Theory,* Academic Press, 1984. [6]

I. Niven and H.S. Zuckerman, *An Introduction to the Theory of Numbers,* 3rd edition, John Wiley & Sons Inc., 1972. [2]

A.W. Nordstrom and J.P. Robinson, An Optimum Non-linear Code, *Information and Control*, 11 (1967), 613-616. [3]

R.E.A.C. Paley, On Orthogonal Matrices, *Journal of Mathematics and Physics*, **12** (1933), 311-320. [2]

F.P. Preparata, A Class of Optimum Non-linear Double-Error-Correcting Codes, *Information and Control*, **13** (1968), 378-400. [3,4]

M. Plotkin, Binary Codes with Specified Minimum Distance, *IEEE Transactions on Information Theory*, 6 (1960), 445-450. [1,2]

K. Sawade, A Hadamard Matrix of Order 268, *Graphs and Combinatorics*, **2** (1986), 185-187. [2]

J. Schönheim, On Linear and Nonlinear Single-Error-Correcting q-nary Perfect Codes, *Information and Control*, 12 (1968), 23-26. [5]

J. Seberry, A Computer Listing of Hadamard Matrices, *Lecture Notes in Mathematics*, N 686 275-281. [2]

N.V. Semakov and V.A. Zinov'ev, Balanced Codes and Tactical Configurations, *Problems of Information Transmission*, **5** (1969) 22-28. [6]

N.V. Semakov, V.A. Zinov'ev, and G.V. Zaitsev, Uniformly Packed Codes, *Problems of Information Transmission*, 7 (1971), 30-39. [6]

N.J.A. Sloane, Recent Bounds for Codes, *Contemporary Mathematics* **9** (1982), 153-185. [7]

N.J.A. Sloane, Private Communication, August 13, 1987. [5]

J.J. Sylvester, Thoughts on Inverse Orthogonal Matrices, Simultaneous Successions, and Tesselated Pavements in Two or More Colors, with Applications to Newton's Rule, Ornamental Tile Work, and the Theory of Numbers, *Philosophy Magazine*, **34** (1897), 461-475. [2]

A. Tietäväinen, On the Existence of Perfect Codes Over Finite Fields, *SIAM Journal of Applied Mathematics*, **24** (1973), 88-96. [5]

R.J. Turyn, Hadamard Matrices, Baumert-Hall Units, Four-Symbol Sequences, Pulse Compressions, and Surface Wave Encodings, *Journal of Combinatorial Theory (A)* **16** (1974), 313-333. [2]

J. H. van Lint, Report of the Discrete Mathematics Group, Report 69-WSK-04 of the Technological University, Eindhoven, Netherlands (1969). [5]

J. L. Vasil'yév, On Nongroup Close-Packed Codes, *Problemy Kibernetiki*, **8** (1962), 337-339. English translation in *Problems in Cybernetics*, **8** (1965), 375-378. [5]

T.J. Wagner, A Remark Concerning the Minimum Distance of Binary Group Codes, *IEEE Transactions on Information Theory*, **11** (1965), 458. [3]

R.M. Wilson, An Existence Theory for Pairwise Balanced Designs, *Journal of Combinatorial Theory*, **13A** (1972) 220-273. [6]

R.M. Wilson, The Necessary Conditions for t-designs are Sufficient for Something, *Utilitas Math.*, **4** (1973) 207-215. [6]

R.M. Wilson, An Existence Theory for Pairwise Balanced Designs: III - Proof of the Existence Conjectures, *Journal of Combinatorial Theory*, **18A** (1975) 71-79. [6]